

CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

150 ptas.

Editorial Delta, S.A.

MI COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen III - Fascículo 30

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones, Francisco Martín
Asesor técnico: Jesús Nebra

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti, A. Cuevas, F. Blasco

Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-94-3 (tomo 3)
84-85822-82-X (obra completa)
Depósito Legal: B. 52/1984

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 088408
Impreso en España - Printed in Spain - Agosto 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.°, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.

Con este fascículo se incluyen las portadillas correspondientes al segundo volumen (fascículos 13-24)



Acceso fácil

El servicio Micronet 800 les permite a los usuarios acceder de manera directa a software del sistema central de Prestel y contar con un punto de intercambio de "correo electrónico"

Ian McKinnell

Micronet 800, servicio que funciona en Gran Bretaña, se llama así en razón de que su sitio original era la página 800 de la base de datos de Prestel, sistema de videotex de la British Telecom. Gracias a él, numerosos usuarios de ordenadores personales pueden tener acceso directo a una considerable variedad de software que puede operarse y, tal vez más importante aún, almacenarse en cinta o disco para su utilización posterior, a menudo gratuitamente. Sin embargo, los abonados necesitan hardware y software adicionales.

Puesto que uno accede a una base de datos remota a través de la red telefónica pública, el primer requisito es un modem (modulador-demodulador), dispositivo de hardware que traduce las señales del ordenador en señales aptas para ser transmitidas telefónicamente. Existen dos alternativas: los modems de cable y los acopladores acústicos. La diferencia básica entre ambos radica en que el modem de cable se conecta al sistema telefónico mediante un enchufe, mientras que el acoplador acústico funciona con el auricular del teléfono (material corriente de la British Telecom), convirtiendo las señales en tonos audibles y viceversa.

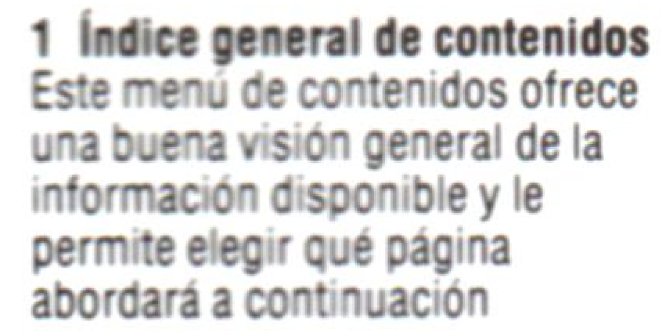
Se puede utilizar todo modem compatible con el Prestel para acceder al Micronet 800, es decir, que

Conexión con el Micronet
Un BBC Micro puede conectarse al Micronet empleando cualquier sistema compatible Prestel. El modo más sencillo de hacerlo consiste en comprar un paquete completo de hardware (un modem o acoplador acústico), software (una ROM, cassette o disco) y abonarse a Micronet/Prestel en Prism Microproducts. En este BBC, la ROM Micronet está situada de modo que el ordenador está preparado para el Micronet en cuanto usted conecta

debe transmitir a 1 200 baudios y recibir a 75 baudios. Existen tres tipos principales de modems. Los modems 1000 y 2000, aptos para la mayoría de los ordenadores personales para los que se dispone de software de Micronet, ofrecen transmisión en dos direcciones (transmisión y recepción en 1 200 baudios, útil para la comunicación entre usuarios de microordenadores así como con Prestel) y la velocidad estándar de 75/1 200 baudios. En el caso del modem 2000, el cambio está controlado con software. El Acoustic Modem, también disponible para diversas máquinas, se limita a operar a 75/1 200 baudios. Su ventaja reside en que no necesita un punto de conexión especial y puede utilizarse con cualquier teléfono, incluso con uno público, siempre, naturalmente, que el pago previo sea lo bastante elevado.

El tercer tipo de modem está hecho a la medida para el Spectrum de Sinclair y se conoce como VTX5000. Al igual que los modems 1000 y 2000, ofrece dúplex completo (transmisiones simultáneas en ambas direcciones) en 75/1 200 baudios y semidúplex (una dirección por vez) en 1 200/1 200 baudios, pero también incluye todo el software necesario para las comunicaciones en forma de ROM, procedimiento mucho más barato que el que se ofrece a los usuarios de otros tipos de microordenadores, que han de comprar software para operar además de hardware adicional.

Dicho software se presenta en dos formas: en



1 Índice general de contenidos
Este menú de contenidos ofrece una buena visión general de la información disponible y le permite elegir qué página abordará a continuación

2 Cartas
Una de las áreas más populares es la página de cartas del usuario; a menudo tanto las cartas como las respuestas de Micronet son un poco crípticas para los novatos

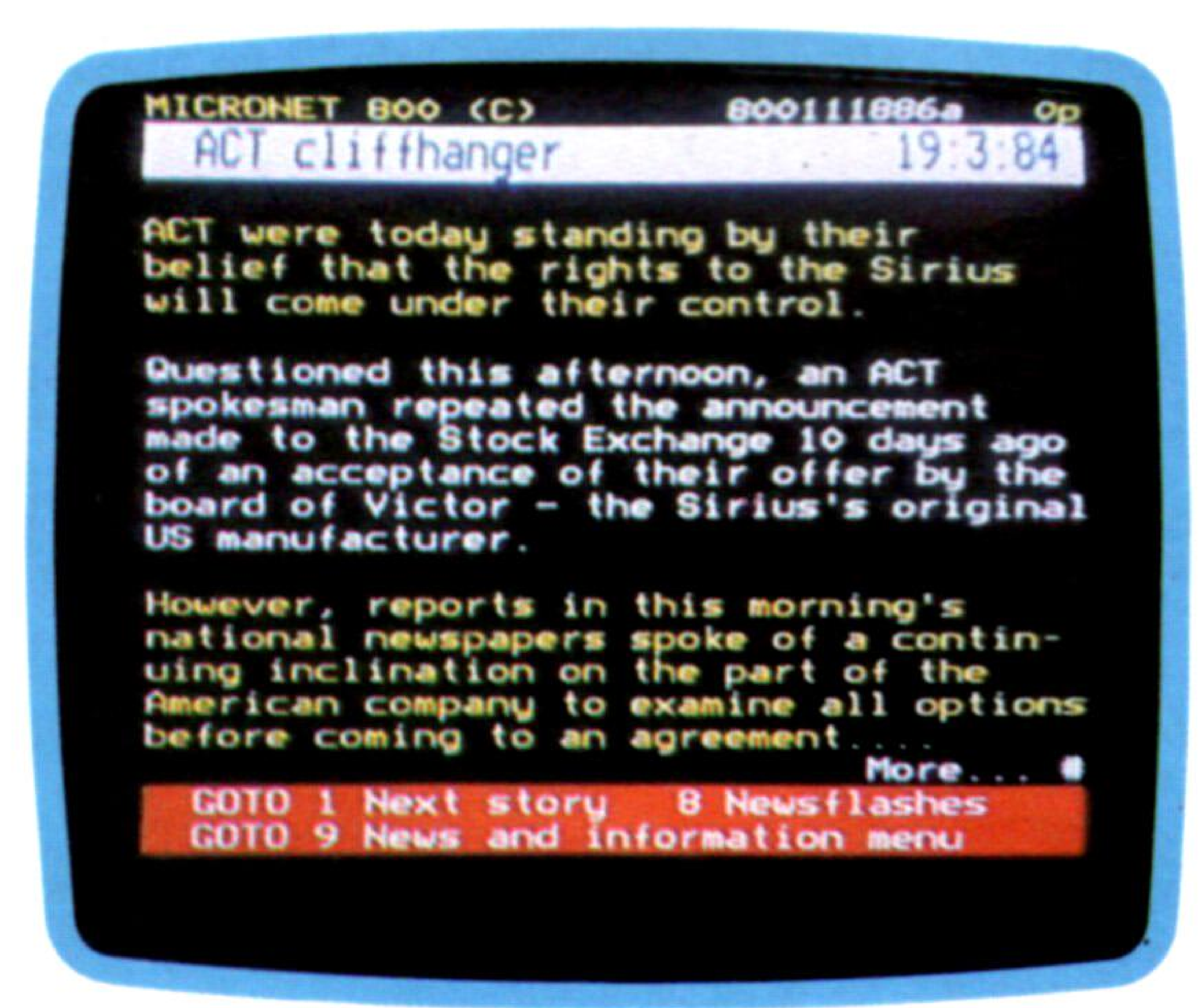




3 Juegos gratuitos
Puede pasar programas de Micronet a su propio ordenador; unos juegos son gratuitos, pero hay otros que se pagan

4 Noticias
Micronet proporciona noticias. Puede actualizar las páginas prácticamente todos los días, por lo que a menudo está adelantado en relación con los semanarios





5 Carlos y Diana
Gallery es una serie de obras maestras de teletexto; también hay juegos, chistes y otros elementos de esparcimiento

6 El tiempo
Abonarse a Micronet le da acceso a muchas otras páginas informativas de Prestel, por ejemplo, la página meteorológica





Liz Heaney

disco o —para el BBC Micro— como ROM sustituible. Los usuarios del BBC pueden optar por la optimización del sistema instalando la ROM en el enchufe de la derecha. En otros ordenadores las funciones son las mismas en cuanto el programa ha sido cargado desde el disco. Puesto que la operación de los sistemas es muy semejante, en líneas generales nos referiremos al Micronet tal como se aplica en el BBC Micro.

El software de decodificación, creado por Scicon —una de las empresas de software más arraigadas en Gran Bretaña—, es necesario para que los microordenadores individuales operen según los protocolos de Prestel, que Micronet utiliza en todo momento (el abonado a Micronet automáticamente cumple los requisitos para acceder a toda la base de datos de Prestel, con su caudal de información sobre todos los temas).

Conexión al sistema

Todos los sistemas de videotex están basados en menús. Al usuario se le presentan una serie de opciones, cada una de las cuales representa una salida de la configuración a la que accede. Cuando el usuario ingresa en el sistema operativo de Micronet, la primera serie de configuraciones solicita códigos de identificación. Se dividen en dos partes y se componen de un identificador personal de diez dígitos y del número de abonado a Prestel. Puesto que Prestel permite que el usuario encargue bienes y servicios, cuyo precio podría ser cargado en su cuenta, dichos números funcionan por derecho propio como tarjeta de crédito. En el caso de Micronet, los números determinan a quién se le cobra el costo del software recibido. Como es lógico, se ha pensado mucho para generar este sistema de seguridad, que parece ser infalible. Por ejemplo, el usuario puede ocultar su número de código de los curiosos incorporando diez asteriscos en su lugar, después de lo cual el sistema le exige que vuelva a incorporarlo, pero esta vez no lo visualiza.

Una vez conectado al sistema, el usuario puede acceder a la base de datos, recibir y enviar mensajes, entregar a voluntad juegos o software comercial... y también recibir, ya que Micronet es un sistema en dos direcciones con alcance no sólo para vender software sino para comprarlo.

Y así ingresa en la base de datos de Micronet propiamente dicha. A fin de ser accesible desde una serie de menús, la estructura de una base de datos ha de ser jerárquica —una disposición tipo "tronco y ramas"—, y Micronet no constituye una excepción a la regla. En cuanto el usuario ha conectado con éxito con el ordenador de Micronet a través del sistema telefónico, la primera opción que se le presenta define el área general en que trabajará: 10 titulares como What's New (Qué hay de nuevo), Computermart (Mercado informático), Talking Back (Réplicas) o Mailbox/Telex (Buzón/Télex). Si bien los títulos por sí solos explican de qué temas se ocupan, existen dos "salidas": una que conduce a la página de Help (Ayuda) y otra que lleva al Prestel propiamente dicho.

Una de las características principales del Micronet consiste en su capacidad para distribuir software. ¿Cómo se cumple esta operación? El menú adecuado se obtiene a través de Telesoftware, y al seguir ese camino se le proporciona al usuario una lista de los tipos de ordenador para los que se dispone de software (cada uno cuenta con su camino de salida propio), así como un menú que ofrece gráficos de los "diez principales" de los ítems más populares, la explicación de cómo vender software a través de Micronet, líneas de Help y un anuncio.

Si seguimos la ruta del BBC Micro, llegamos a una configuración que permite elegir entre paquetes individuales de software (juegos, paquetes de gestión, programas de servicios), últimas novedades, *bestsellers*, programas gratuitos y configuraciones Help. Al seleccionar una de las cuatro opciones que ofrecen programas aparece una lista de títulos con breves descripciones. La elección de uno de ellos lleva a Micronet a visualizar la configuración conocida como Downloader Menu (menú de compra). Esto define las opciones disponibles, que consisten en recibir y utilizar el programa, desconectar el sistema o hacerlo funcionar y continuar conectado al Micronet. Permanecer conectado puede resultar caro desde el punto de vista de las tarifas telefónicas en el caso de que haga usted algo más que guardar en cinta o disco el programa que acaba de adquirir.

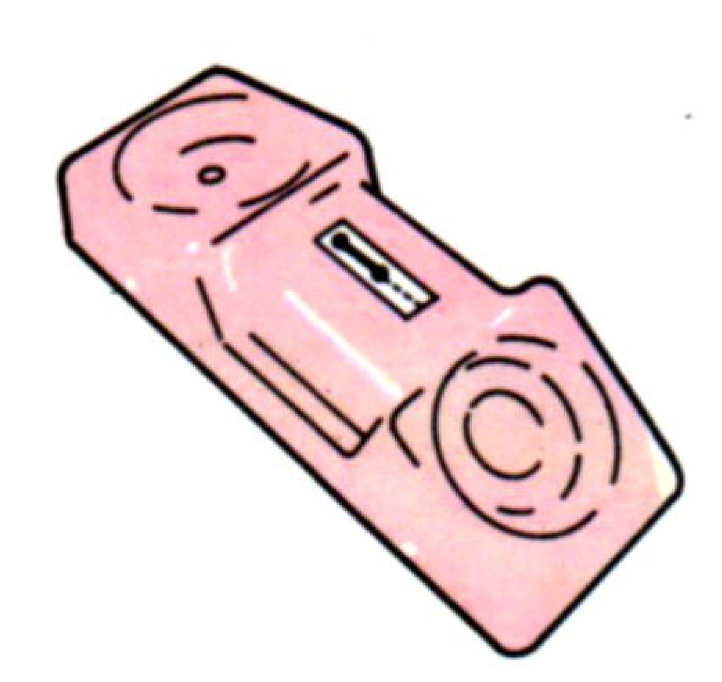
Aunque es verdad que Micronet ofrece software gratuito, la cantidad es limitada, a pesar de que Micronet sostiene que dispone de más de cien programas en cualquier momento dado. Muchos de los programas gratuitos son producciones de aficionados, por lo que no siempre puede esperarse sonido y símbolos gráficos de alta calidad. Los demás programas disponibles tienen diversas tarifas, ya que el precio lo fija el autor.

Puesto que hay que pagar gran parte del software, quizá debería contemplarse este aspecto de Micronet como un sistema alternativo de distribución de programas más que como un servicio público. A decir verdad, los esfuerzos de Prism por encontrar un método de distribución más eficaz no se agotan en Micronet: la empresa es el representante británico del sistema norteamericano Romox, que programa cartuchos de ROM especialmente destinados a la venta al por menor.

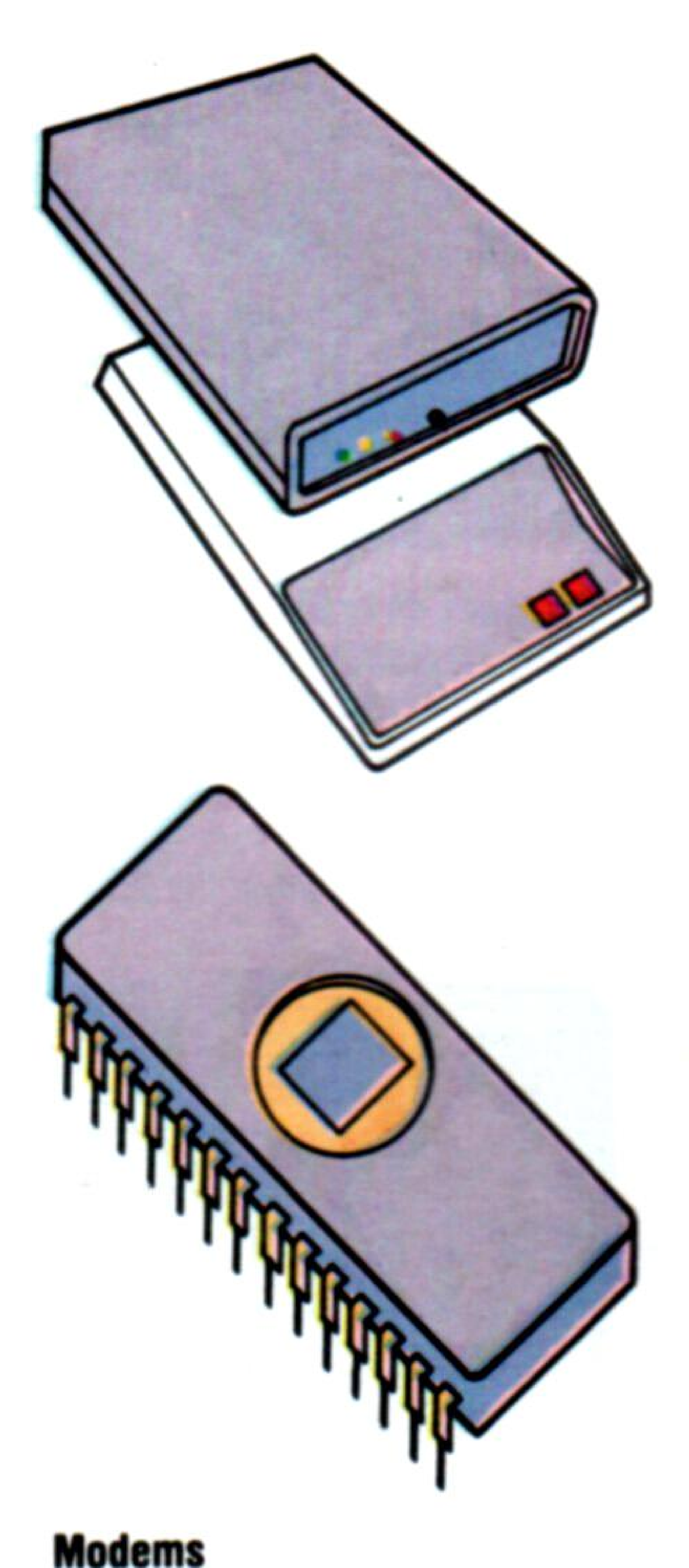
Las restantes funciones del Micronet son semejantes a las del sistema Prestel, del que forma parte: y así, Mailbox se encuentra reproducido en el sistema mayor, como las páginas de noticias y anuncios.

El camino a través de la jerarquía, tal como lo determina la salida escogida por el usuario, se define por un número de identificación de la configuración. Si se conoce su número de identificación, es posible evitar la en ocasiones larga ruta de acceso secuencial conectando directamente cualquier configuración. Se trata de una característica útil para el usuario con experiencia, que probablemente se sentirá frustrado al tener que verse obligado a recorrer página tras página de una información que no le interesa.

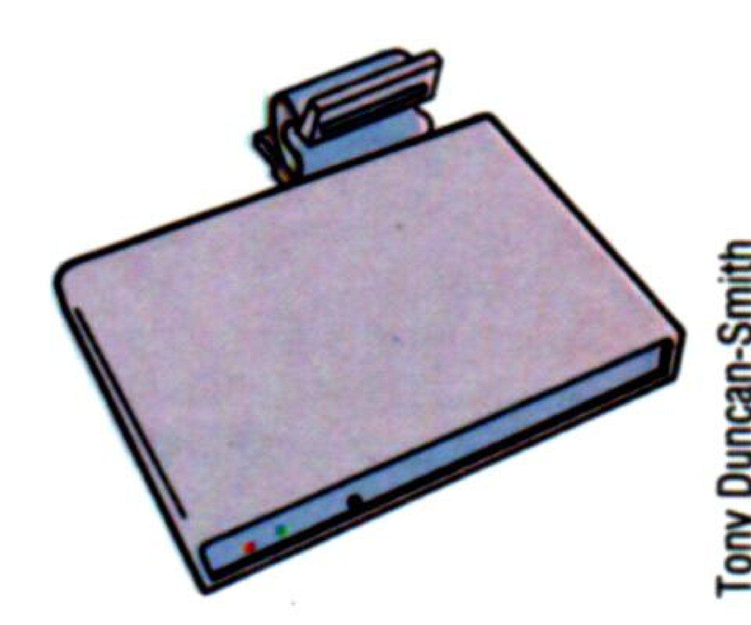
Si tenemos en cuenta la cantidad y calidad de los servicios que ofrece, el Micronet no es caro. La suscripción cuesta 1 libra semanal a los usuarios privados que operan desde sus casas y algo menos de 2 libras a los usuarios comerciales y a las escuelas. A ello se añade el costo del modem y, desde luego, la llamada telefónica local que conecta al usuario con el ordenador de Micronet o Prestel... más el costo adicional que se deba pagar por cualquier programa que uno compre.



Acopladores acústicos
El acoplador acústico ofrece el modo más sencillo de conectar máquinas como el BBC, el RML 380Z y el Apple a Micronet. Necesitará el software adecuado para su máquina, si bien la lista de ordenadores a los que se puede tener acceso con este software crece constantemente



Modems
Los Modems 1000 y 2000 se conectan directamente con la red telefónica y por ende ofrecen una vía de acceso más fiable al Micronet. El software adecuado para el BBC puede ofrecerse en un chip de ROM, de modo que está disponible desde el instante mismo en que lo conecta



Tony Duncan-Smith

Para el Spectrum
El VTX 5000 es un modem diseñado específicamente para el Spectrum y se conecta al conector de ampliación

Ruedas de fuego

Con la introducción de la unidad de disco para sus modelos 32 y 64, Dragon se mantiene a la altura de sus más serios competidores

Las unidades Dragon emplean discos de densidad única de 13,3 cm. Se forman magnéticamente, según órdenes, con 40 pistas divididas en 18 sectores, cada uno de los cuales posee un espacio de almacenamiento de 256 bytes. En consecuencia, la capacidad total de almacenamiento por disco es de 40 × 18 × 256 = 184 320 bytes o 180 Kbytes. El DOS (*Disk Operating System*: sistema operativo de disco) necesita parte del espacio de almacenamiento para identificación de archivo, manejo de disco y directorio, lo que deja alrededor de 175 Kbytes de espacio formateado.

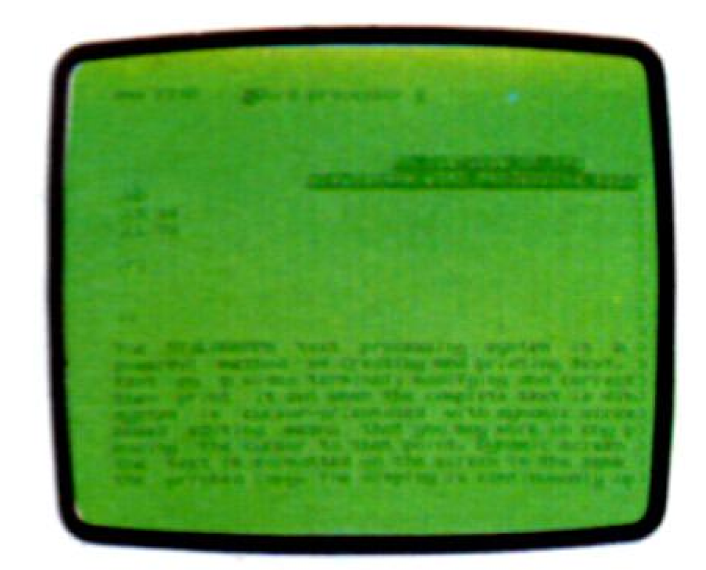
El directorio consiste en la lista de todos los archivos contenidos en un disco. Según instrucciones, visualiza los nombres de los archivos, los tipos de archivo, la cantidad de bytes del archivo y el número de bytes disponibles. Pueden existir cuatro tipos de archivo: programa BASIC, de datos, binario y de seguridad (*backup*). El tipo de fichero almacenado se indica añadiendo .BAS, .DAT, .BIN y .BAK al nombre correspondiente a cada fichero. El DOS de Dragon no menciona de qué forma los archivos de un disco se distribuyen en la memoria y cómo se accede a ellos pero, puesto que no hay órdenes para cerrar el espacio libre de un disco, probablemente existe una forma de BAM (*Block Availability Map*: mapa de disponibilidad de bloques).

Aunque parezca extraño, el DOS de Dragon no puede acceder aleatoriamente a los datos, de un byte por vez. Éste es un medio útil para un programa que requiere leer a menudo pequeñas cantidades de datos, por ejemplo, una base de datos. En su lugar, el DOS utiliza lo que se denomina *acceso directo simulado*, que aparentemente hace lo mismo, salvo que se requieren unas pocas líneas adicionales de código en BASIC para ejecutarlo.

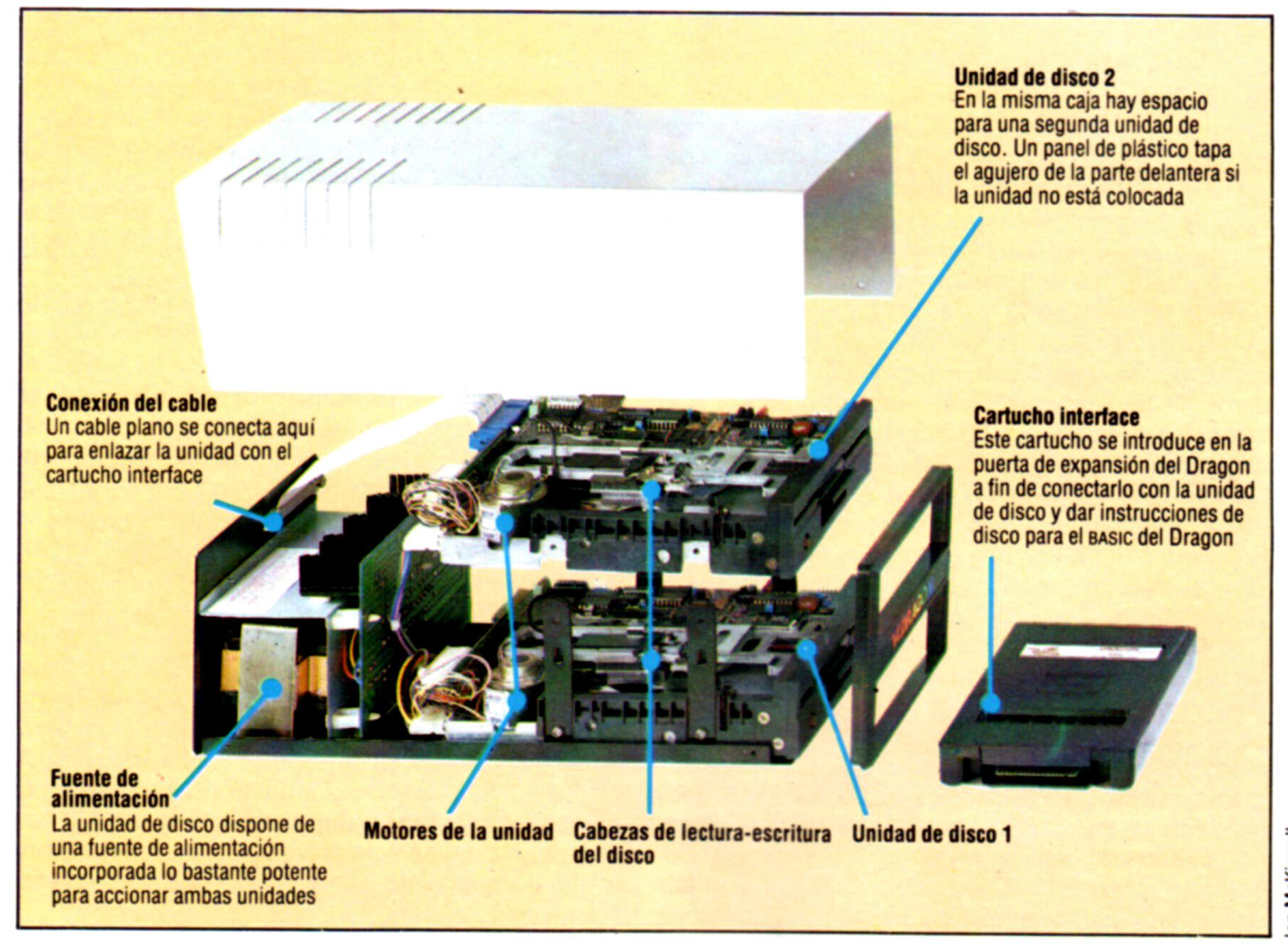
El precio relativamente elevado y la reducida capacidad de almacenamiento hacen que este sistema de disco sólo compense parcialmente su costo. Tal vez consciente de ello, Dragon Data piensa incrementar su variedad de unidades a fin de incluir, a un precio razonable, unidades de 80 pistas, distribuidas en dos lados, y doble densidad. Para el Dragon 64 se está introduciendo como opción un sistema operativo de RAM residente cargado a partir del disco. Denominado 0S9, requiere 16 Kbytes de RAM, pero cuenta con un avanzado sistema operativo 6809 que incluye un DOS excepcionalmente eficaz, y en discos separados dispone de PASCAL, C, COBOL y BASIC 09 estructurado.



Opciones de lenguaje
El primer grupo de software de disco de Dragon funciona según el sistema operativo 0S9 e incluye lenguajes informáticos avanzados como el PASCAL y el compilador C, que vemos aquí



Paquete sofisticado
El 0S9 también permite utilizar sofisticados programas de oficina. Este paquete de tratamiento de textos incluye un verificador de ortografía con un diccionario de 42 000 palabras



Fuerza motriz
Las unidades de disco Dragon se presentan en una caja con su fuente de alimentación incorporada y espacio para dos unidades. Un cable plano la conecta a un cartucho interface que luego se enchufa en el Dragon. Los discos funcionan según el sistema operativo Dragon o el más costoso y profesional tipo software 0S9. Cada disco puede albergar 175 K de información



Órdenes del Dragon

Cuando se escriben órdenes, los parámetros que describen el archivo suelen ser los mismos. En este caso, el formato de instrucción utilizado con más frecuencia es:

COMMAND"D:NOMARCHI.TYP"

donde D selecciona una unidad (1-4) y hace caso omiso de la unidad omitida; NOMARCHI consta de hasta ocho caracteres que especifican el archivo, y .TYP es el código de identificación de éste, que puede ser especificado por el usuario; si no lo es, el DOS asume el valor omitido: .BAS. Este formato está representado por FSP.

DRIVE N
En este caso, N puede ser de 1 a 4. Esto selecciona la unidad omitida.

DSKINIT
Esta orden formatea el disco especificado:

DSKINIT D,S,T

En este caso D selecciona la unidad; S el lado del disco (ya sea 1 o 2) a ser formateado, siendo 1 el omitido; T selecciona el número de pistas a formatear (40 u 80), siendo 40 el omitido. Con un sistema estándar de una sola unidad, basta con teclear DSKINIT (ENTER) y el disco insertado se borrará y se formateará correctamente.

DIR D
Esta orden visualiza el directorio de la unidad especificado, D, del siguiente modo:

```
DIR
PROGNAME .BAS 1654
M/C BINARY .BIN 1389
PROGDATA .DAT 2581
PROGNAME .BAK 1654
167322 FREE BYTES
```

ARCHIVOS DE PROGRAMAS

SAVE
Esta orden almacenará programas o archivos binarios en disco. SAVE FSP almacena un archivo de programas. No hace falta especificar .BAS ya que se trata del valor omitido. La orden:

SAVE FSP,CCCC,FFFF,XXXX

almacena un archivo binario. En este caso CCCC es la dirección de inicio en decimal del código a almacenar; FFFF es la dirección final y XXXX es la dirección a partir de la cual se ejecuta el programa.

LOAD
Leerá el programa o archivos binarios del disco con la orden:

LAOD FSP

Si el archivo especificado es binario, FSP puede suplirse por, CCCC. Ello indicará la nueva posición de inicio en la memoria del archivo binario.

RUN FSP
Inmediatamente cargará y ejecutará el programa en BASIC especificado.

CHAIN FSP
Cargará y ejecutará un programa sin modificar las variables almacenadas. Resulta útil en los programas que comparten información. Añadir #,CCCC tendrá el mismo efecto que en la orden LOAD descrita.

FREE D
Visualiza la cantidad de bytes libres en una unidad específica.

COPY
Duplica el archivo ARANTIG como ARNUEVO del siguiente modo:

COPY ARANTING TO ARNUEVO

Si los números de la unidad no se especifican, la copia se hace en el mismo disco en la unidad omitida.

RENAME
Cambia el nombre pero no el tipo de un archivo. Ambos FSP deben referirse a la misma unidad o asume el omitido

MERGE FSP
Esta orden superpondrá un archivo de programas en BASIC específico en otro contenido en la memoria, haciendo así que los programas se fundan. El programa contenido en el disco tendrá preferencia si los números de línea coinciden.

KILL FSP
Borrará un archivo específico del disco.

PROTECT
Esta orden evita que un archivo sea borrado o que se le escriba encima mediante cualquier otra instrucción salvo DSKINIT:

PROTECT ON FSP

También hará que se visualice un campo P invertido con el nombre del archivo en el directorio del disco. La orden PROTECT OFF FSP retirará la protección

BACKUP
Copiará todo el contenido de un disco en una unidad (DA), pasándolo a un disco de otra (DB) con la instrucción:

BACKUP DA TO DB,S,T

S y T tienen exactamente los mismos valores que para la orden DSKINIT, permitiendo hacer copias en discos y unidades de distinto formato. El usuario de discos simples puede limitarse a digitar BACKUP (ENTER), después de lo cual se visualizan las instrucciones para alternar discos de fuente y de destino.

VERIFY
ON y OFF controlan esta orden, que comprueba que el contenido de un archivo almacenado es igual al original.

ARCHIVOS DE DATOS

FWRITE
Se utiliza para crear y escribir un archivo de datos que contenga listas de variables. Para cada FWRITE se abre un canal al archivo, permitiendo añadir nuevos datos a los ya almacenados. Se escribe así:

FWRITE"NOMARCHI";VAR

En este caso NOMARCHI es el archivo a escribir, o a crear y escribir, y VAR es una lista de variables que contiene los datos a almacenar. Comas y dos puntos ponen fin a las cadenas de variables en los archivos de datos, a menos que se tenga el propósito de que sean leídas como cadenas por FLREAD. La orden:

FWRITE"NOMARCHI",FROM SB,FOR TB;VAR

escribe VAR al NOMARCHI comenzando en el byte SB, extendiendo la longitud de la lista hasta un total de bytes TB. Simultáneamente sólo pueden estar abiertos diez archivos a los que se accede por FWRITE.

CLOSE D
Cierra los canales de los archivos abiertos por FWRITE, FREAD y FLREAD.

CREATE"NOMARCHI",LA
Crea un archivo NOMARCHI, de LA bytes.

FREAD
Esta orden se construye como FWRITE. VAR es leído en la memoria o, si tomamos el segundo ejemplo dado, VAR es leído a partir del byte SB. La aguja de lectura es adelantada entonces por los bytes SB+TB.

FLREAD
Se forma como FREAD, pero comas y dos puntos no se leen como terminaciones.

EOF
Se utiliza para indicar la última entrada válida en un archivo que se está leyendo. Ej.:

EP = EOF ("NOMARCHI")

donde EP es 0 hasta que el puntero de lectura lee el registro final y cambia EP a 1.

LOC"NOMARCHI"
Mostrará la posición del puntero de lectura como el número del siguiente byte a leer de un archivo específico.

SWRITE
Almacenará datos en el sector S de la pista T como las cadenas A$ y B$ hasta un máximo de 128 bytes cada una, así:

SWRITE D,T,S,A$,B$

SREAD
Recuperará los datos almacenados por SREAD empleando el mismo formato. A A$ y B$ se les pueden adjudicar diferentes nombres de variables.

Caminos tortuosos

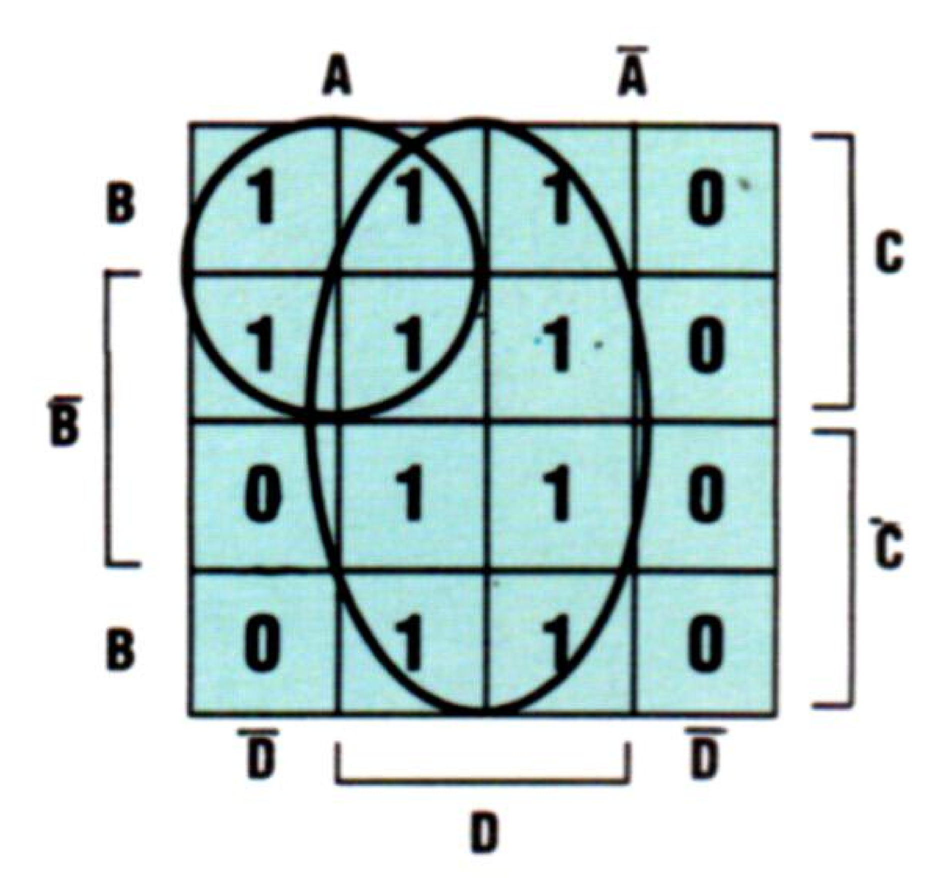
Examinaremos en el presente capítulo los diagramas de Karnaugh cuando se usan para expresiones booleanas con cuatro variables y cómo se aplican en el diseño de circuitos

Cuatro variables. En los casos de expresiones booleanas que suponen cuatro variables, los diagramas de Karnaugh —y las expresiones mismas— pueden parecer sumamente difíciles. Pero si aplicamos las ideas sencillas que establecimos al estudiar los diagramas de Karnaugh de dos y tres variables, pronto comprobaremos cómo se nos hacen familiares y se convierten en fáciles de manipular.

Supongamos, por ejemplo, que nos piden que simplifiquemos la siguiente expresión:

$$ABC\overline{D} + ABCD + \overline{A}BCD + A\overline{B}C\overline{D} + \overline{B}CD + A\overline{B}\overline{C}D + \overline{A}\overline{B}\overline{C}D + B\overline{C}D$$

Vemos que se necesita un diagrama de Karnaugh de cuatro variables y que, si bien esta expresión consta de ocho componentes, en realidad necesitaremos llenar 10 del diagrama (los términos $\overline{B}CD$ y $B\overline{C}D$ representan dos casos cada uno). Así, el diagrama es:



En el diagrama vemos que el grupo central de ocho unos representa todas las combinaciones posibles que contienen la D. El grupo de cuatro unos en el extremo superior izquierdo abarca todos los casos posibles en que están incluidas A y C. En consecuencia, la expresión se simplifica en A AND C OR D (expresada así: A.C + D).

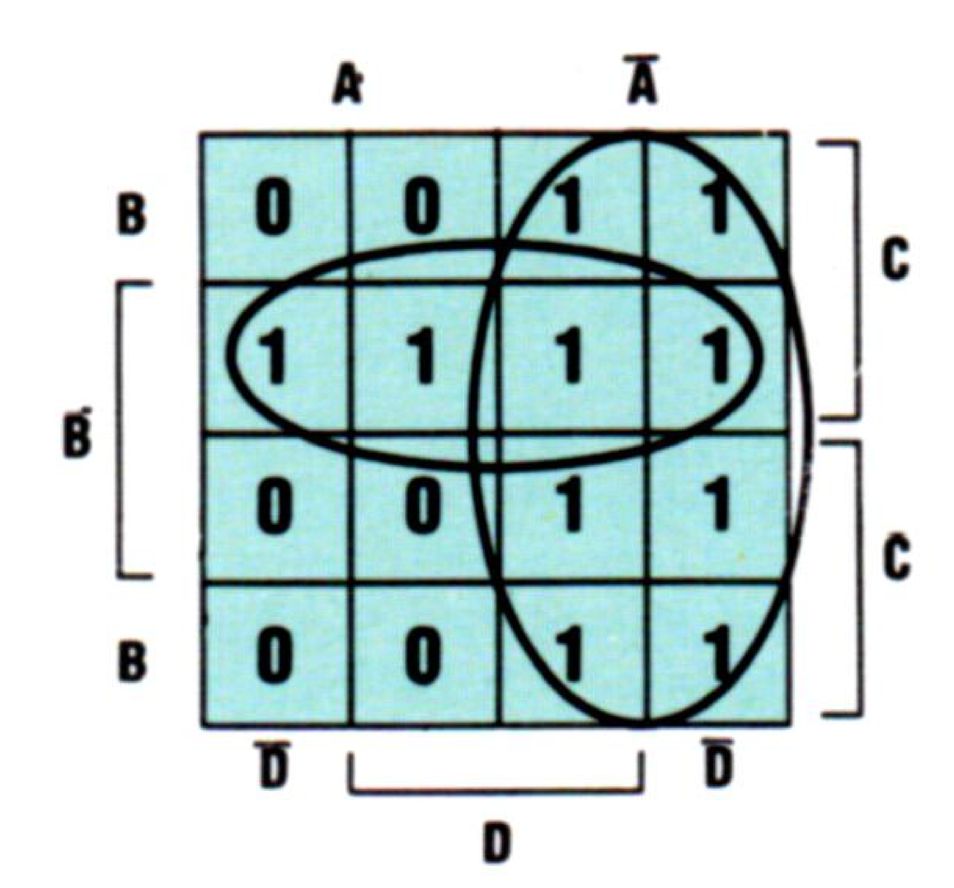
En ocasiones es necesario transformar una expresión a fin de que pueda ser representada en un diagrama de Karnaugh, como en el siguiente ejemplo:

$$\overline{A + B + \overline{C}} + \overline{A}.B + \overline{B + \overline{\overline{C}}}$$

En este caso primero debemos aplicar la ley de Morgan (véase p. 526) antes de dibujar el diagrama correspondiente. Se reformula así:

$$\overline{A}.\overline{B}.\overline{C} + \overline{A}.B + \overline{B}.C$$

y el diagrama producido por esta expresión es el que se representa a continuación:



En el diagrama de Karnaugh se ve que esta expresión queda simplificada en: $\overline{A} + \overline{B}.C$. Recurriendo una vez más a la ley de Morgan, la expresión queda finalmente simplificada como:

$$\overline{A.(B + \overline{C})}$$

Diseño de circuitos

Ejemplo 1: Treinta días

Probablemente conoce la cantinela: "Treinta días trae noviembre con abril, junio y septiembre...". Supongamos que cada mes del año está codificado en un código binario de cuatro bits: 0001 para enero hasta el 1100 para diciembre. Nuestra tarea consiste en diseñar un circuito que acepte el código de cuatro bits como entrada y un 1 de salida si la entrada del mes tiene treinta días.

La tabla de verdad de dicho circuito queda así:

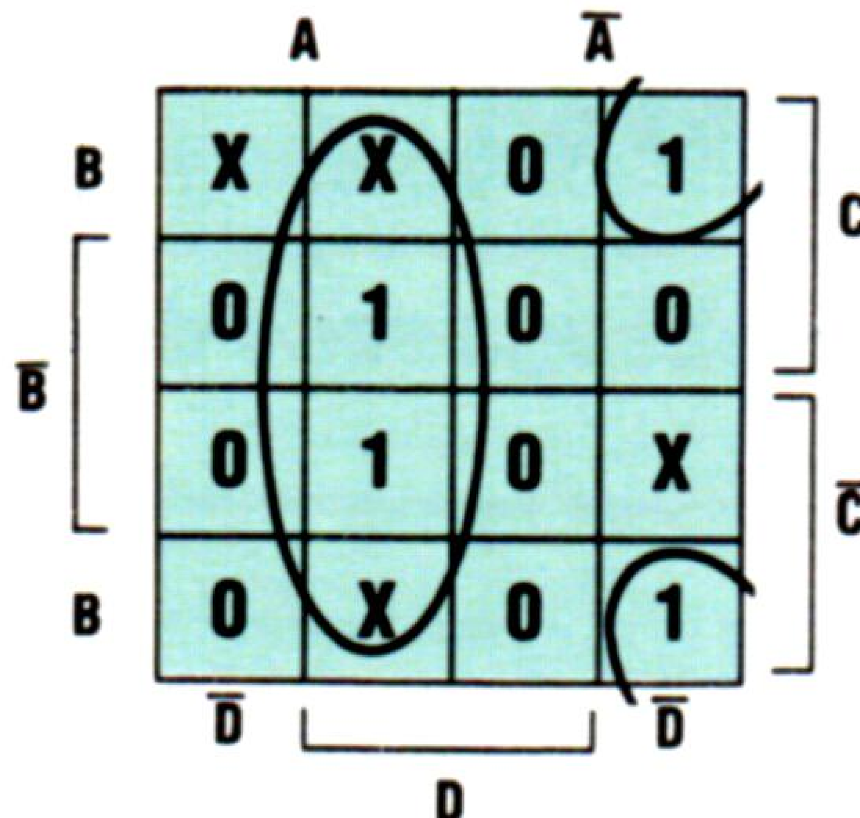
MES	ENTRADAS				SALIDA
	A	B	C	D	S
	0	0	0	0	X
ENE	0	0	0	1	0
FEB	0	0	1	0	0
MAR	0	0	1	1	0
ABR	0	1	0	0	1
MAY	0	1	0	1	0
JUN	0	1	1	0	1
JUL	0	1	1	1	0
AGO	1	0	0	0	0
SEP	1	0	0	1	1
OCT	1	0	1	0	0
NOV	1	0	1	1	1
DIC	1	1	0	0	0
	1	1	0	1	X
	1	1	1	0	X
	1	1	1	1	X

La salida X de la tabla de verdad significa una entrada nula. Supondremos que el circuito *no* recibirá esas señales. A partir de la tabla de verdad, para S = 1, podremos formar la siguiente expresión booleana con los bits binarios de la entrada:



$S = \overline{A}.B.\overline{C}.\overline{D} + \overline{A}.B.C.\overline{D} + A.\overline{B}.\overline{C}.D + A.\overline{B}.C.D$

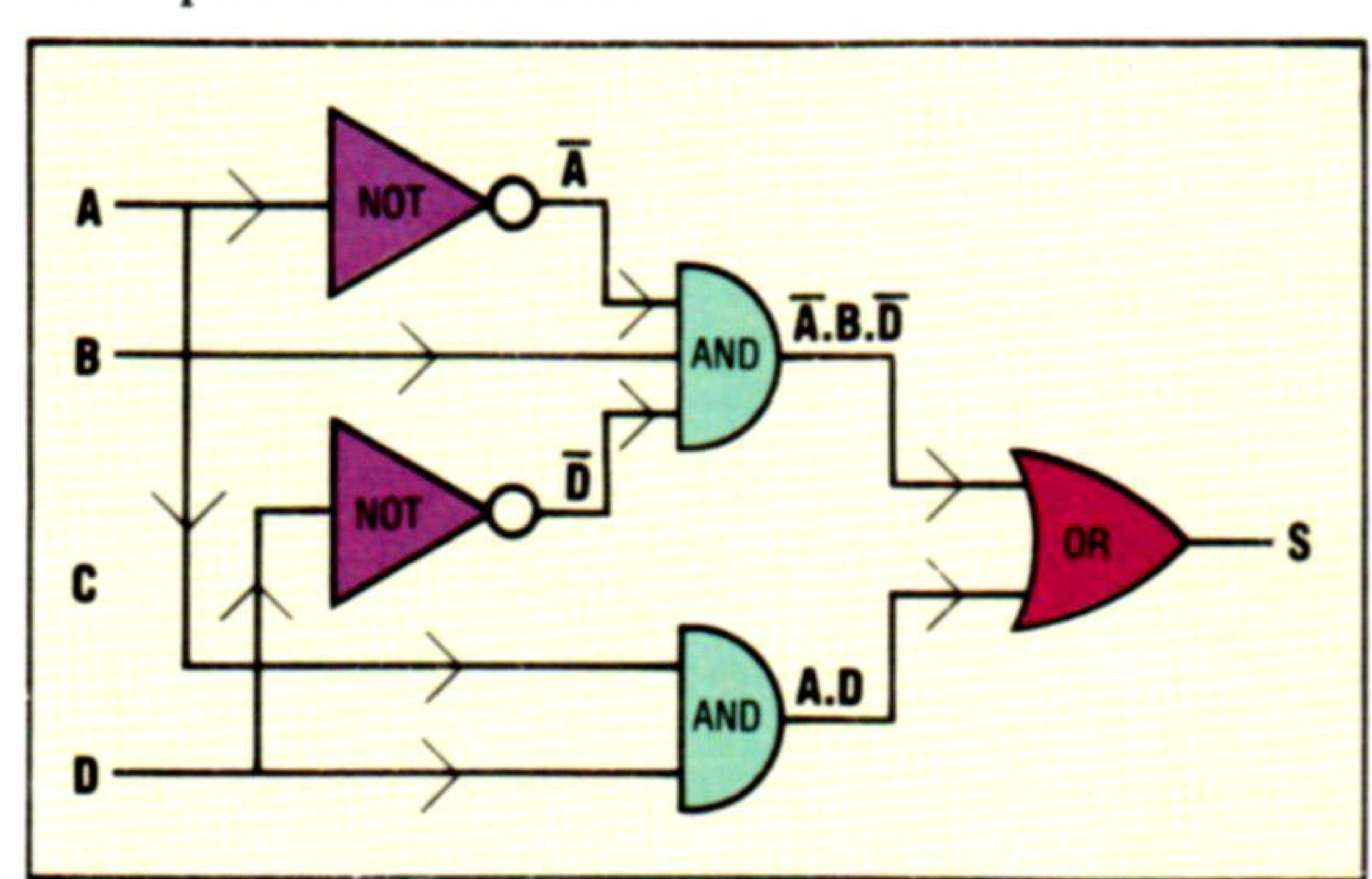
Al dibujarla en un diagrama k junto con las condiciones de "entrada nula" (X), tenemos:



Con este diagrama vemos que la expresión se reduce a:

$A.D. + \overline{A}.B.\overline{D}$

Así, nuestro circuito de señales del "mes de treinta días" puede construirse:



Ejemplo 2: Números impares

Puesto que los números del 0 al 15 pueden codificarse mediante cuatro dígitos binarios (del 0000 al 1111), se nos pide que diseñemos un circuito que acepte el código de cuatro bits como entrada y cuya salida dé un 1 si la misma representa un número impar mayor que dos.

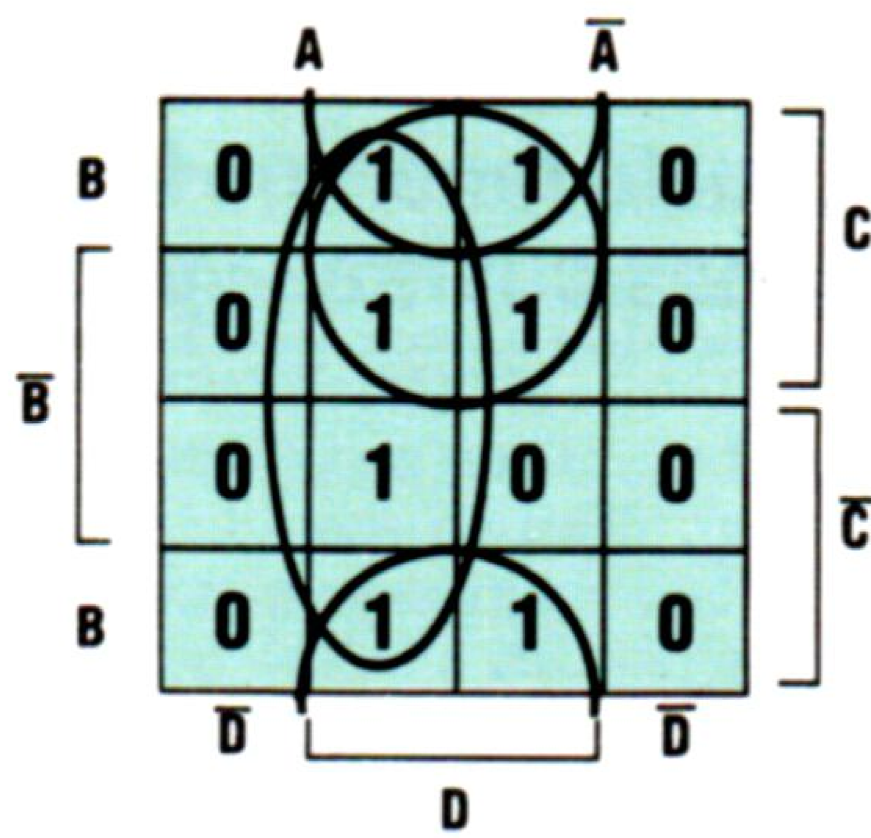
Haremos primero la tabla de verdad:

NÚMERO DECIMAL	ENTRADAS A	B	C	D	SALIDA S
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

A partir de esta tabla de verdad obtenemos la siguiente expresión de álgebra booleana, para todas las condiciones donde S es verdad (= 1):

$S = \overline{A}.\overline{B}.C.D + \overline{A}.B.\overline{C}.D + \overline{A}.B.C.D + A.\overline{B}.\overline{C}.D + A.\overline{B}.C.D + A.B.\overline{C}.D + A.B.C.D$

El diagrama de Karnaugh de esta expresión es:



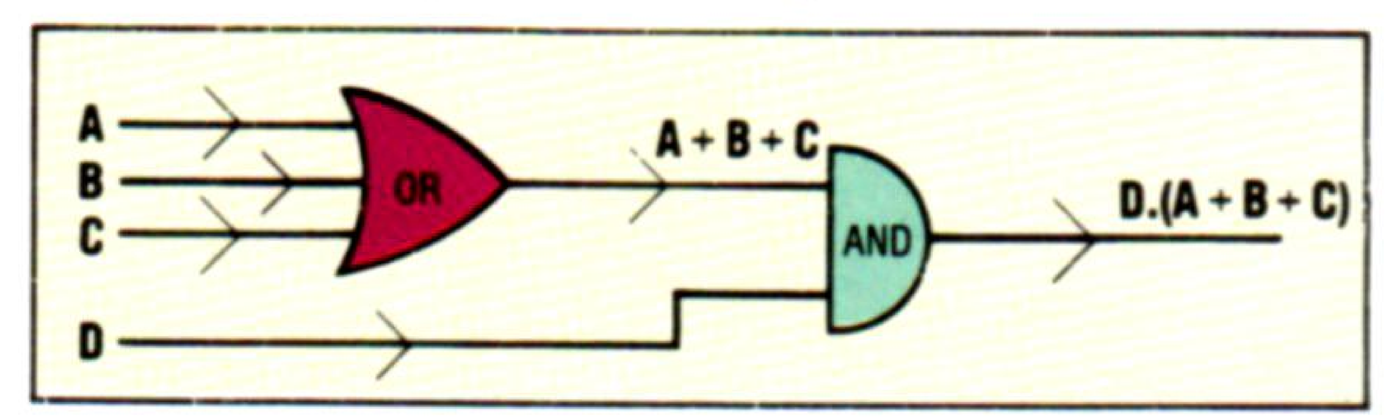
A partir del diagrama, es posible emplear tres grupos de cuatros, representados así:

$S = A.D + C.D + B.D$

y puede simplificarse aún más utilizando la propiedad distributiva hasta obtener:

$S = D.(A + B + C)$

En consecuencia, el circuito puede diseñarse:



En el próximo capítulo repasaremos los aspectos más importantes del curso de lógica y ofreceremos un amplio conjunto de ejercicios de repaso.

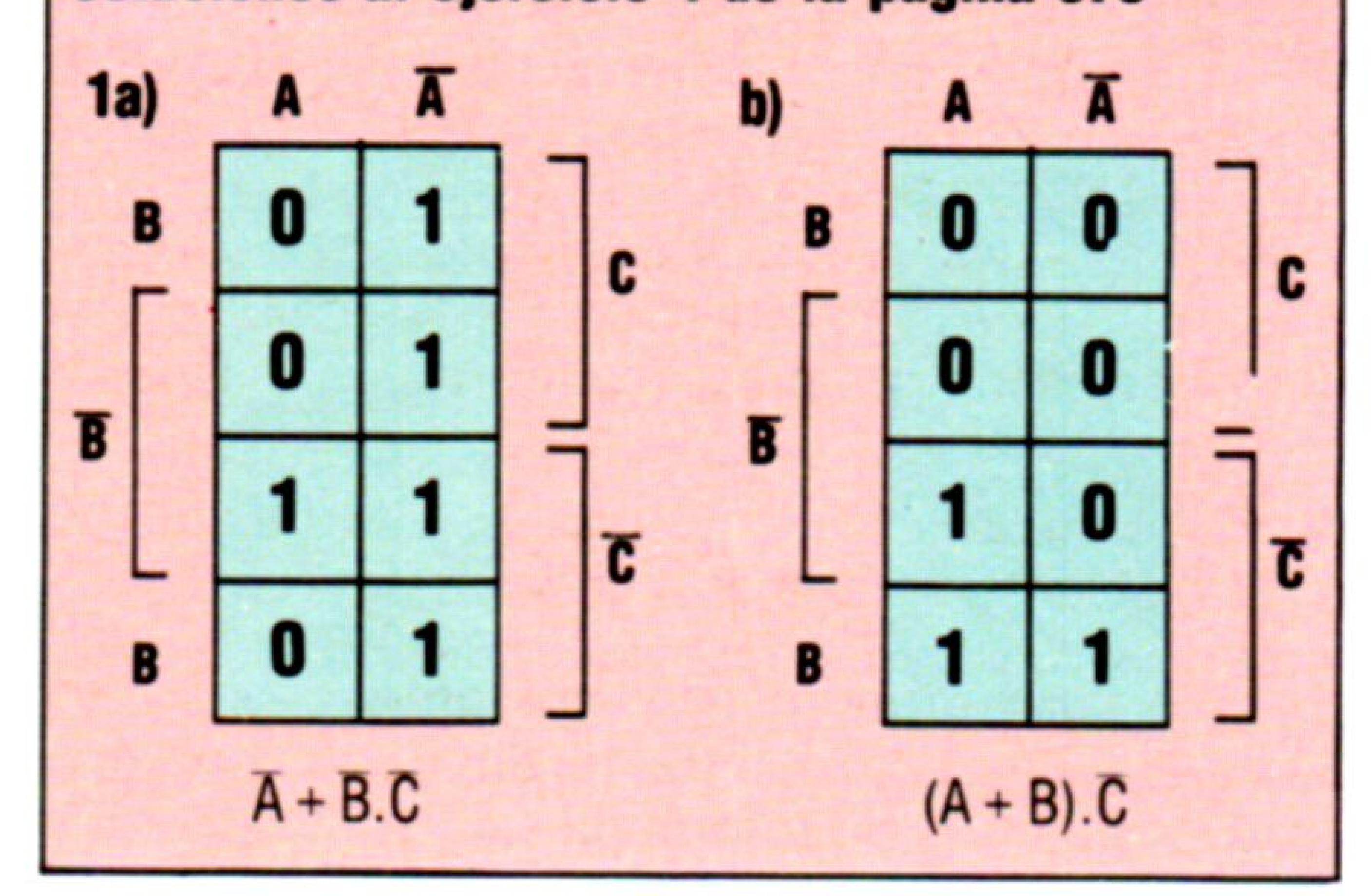
Ejercicio 5

1) Simplifique las siguientes expresiones de álgebra booleana utilizando diagramas de Karnaugh:

a) $A.B.C + A.\overline{B}.\overline{C} + \overline{A}\overline{C} + \overline{A}\overline{B}.C + A.B.\overline{C}$
b) $\overline{B + C} + B.\overline{C} + A.C$
c) $A.\overline{B}.D + \overline{A}.D + A.B.C.D + A.B.\overline{C} + \overline{A}.B.\overline{C}.\overline{D}$

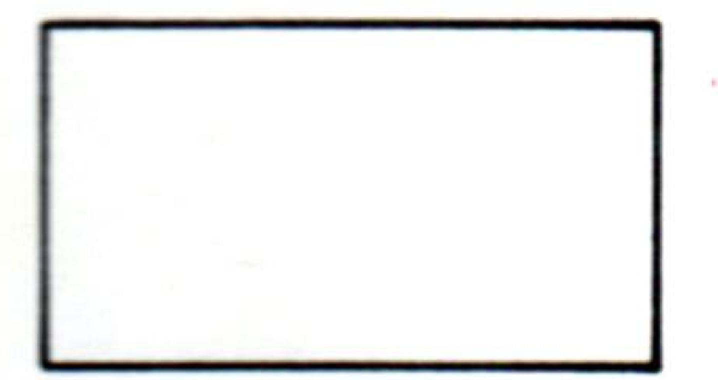
2) Diseñar un circuito que acepte las representaciones binarias de los números enteros entre 0 y 7, ambos inclusive. El circuito ha de dar una salida si la entrada numérica es impar o múltiplo de 3 (p. ej., 3 o 6). Dibuje una tabla de verdad y obtenga la expresión simplificada; dibuje un circuito lógico que lleve a cabo esa función.

Soluciones al ejercicio 4 de la página 573



Bucles incondicionados

Con un bucle tenemos la posibilidad de repetir una serie de operaciones cuantas veces deseemos



Operación
Con este símbolo se representa una elaboración, o sea la ejecución de una operación o de un conjunto de ellas, tales como cálculos aritméticos, cambios o sustituciones de valores, modificación de datos o de su posición, etc.

Veamos ahora un ejemplo con una operación aritmética. Se nos entregan tres fichas; cada una contiene una cantidad. Debe visualizarse el total de la suma de dichas cantidades y también el cuadro del contenido de cada ficha.

En primer lugar, puede verse que se necesitan una serie de campos destinados a contener los datos que se deben visualizar como final del proceso. En el ejemplo de la figura 1 se han empleado los siguientes: TOTAL para designar el campo en el que se van sumando las diferentes cantidades y C1, C2 y C3 para indicar los cuadrados de las fichas correspondientes.

Así, el proceso queda como sigue: leer una ficha, la cantidad que en ella figura se acumula en el campo TOTAL; después se calcula el cuadrado de dicha cantidad, que queda guardado en el campo correspondiente, y una vez realizadas estas operaciones con las tres fichas, como final de proceso, se visualizan los campos que guardan los tres cuadrados y el de la suma, que contará en ese momento con la suma de las tres fichas.

Este desarrollo del proceso se podría considerar válido en ejemplos tan sencillos como éste, en que se cuenta con un número reducido de fichas, pero supongamos que hemos de trabajar con un número superior de fichas (50 o 100). Entonces la labor, además de extensa, sería reiterativa, ya que se observa que una misma parte del esquema se repite: precisamente, aquella que corresponde a la lectura de la ficha, a la suma de los valores y al cálculo del cuadrado.

La figura 2 nos muestra cómo reproducir una misma parte del proceso. Tras la entrada de la cantidad perteneciente a una ficha, se realiza la suma de dicha cantidad en el campo TOTAL, procediéndose después a averiguar el cuadrado y a visualizar ambos campos. Acto seguido, la línea de flujo retorna al punto inicial, en el que el proceso comienza de nuevo. Ahora bien, en dicho proceso no hay un final lógico, con lo que se obtendrá una repetición ilimitada, ya que al no alcanzar jamás un final, se convertirá en un círculo cerrado en cuyo interior se realiza una parte del proceso. Esta forma de repetición se denomina *bucle incondicionado*. Este nombre se explica por el hecho de que cuando se alcanza el punto en que aparece la línea conductora, el flujo automáticamente sigue su dirección, con lo que se convierte en un reciclaje infinito. Sólo podrá detenerse por la intervención del operador mediante una ruptura voluntaria del ciclo. Esta ruptura será objeto de una próxima lección.

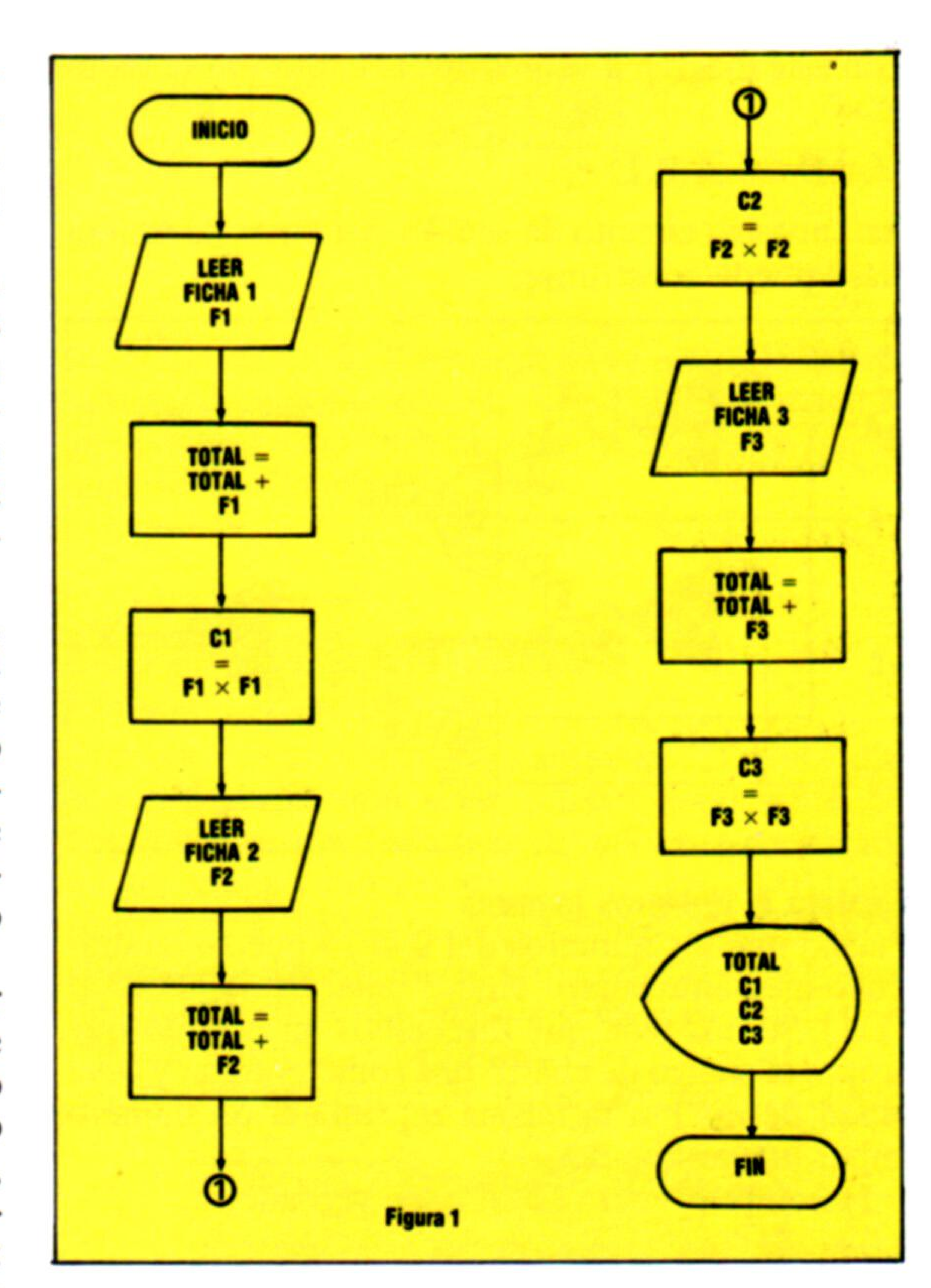


Figura 1

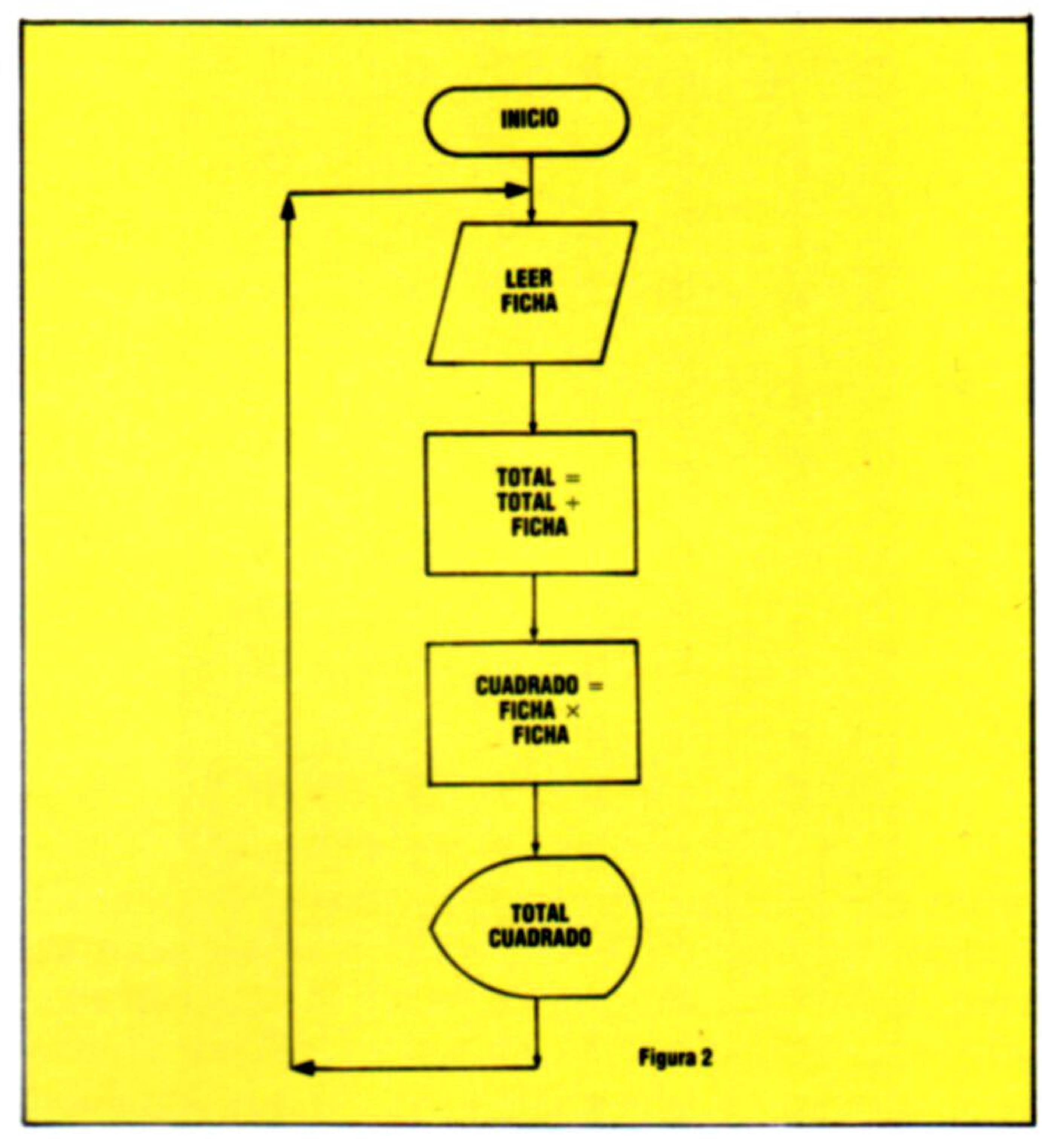


Figura 2

Un diseño austero

El AIM 65 es un microprocesador creado especialmente para aumentar sus conocimientos acerca de cómo funcionan los ordenadores

No todos los microordenadores se alojan en carcasas vistosas, cuidadosamente diseñadas. Diversas máquinas, desde las más sencillas hasta las más sofisticadas, se venden como tableros al desnudo. Uno de los más polifacéticos es el Advanced Interactive Microcomputer, fabricado por la firma Rockwell, más conocido como AIM 65 y concebido como ayuda educativa y de experimentación.

En su presentación más sencilla, el AIM 65 es un tablero al desnudo, desprovisto de carcasa. No obstante, desde el punto de vista operativo, es muy versátil y constituye la única máquina experimental que lleva incorporada una impresora. La mayoría de los sistemas experimentales carecen de un teclado propiamente dicho y no poseen la útil visualización LED de 20 caracteres y 16 segmentos que el AIM proporciona al usuario.

Si bien estas tres características no son decisivas, las máquinas más corrientes —con teclado hexadecimal, visualización de ocho caracteres y siete segmentos y sin impresora— resultan mucho más difíciles de operar. Los requisitos básicos son canales de input/output y memoria, y en este aspecto el AIM es más típico con sus cuatro Kbytes estándar de RAM y 12 Kbytes de ROM; sin embargo, en este sentido está mejor dotado que otras máquinas experimentales, que sólo cuentan con un Kbyte de RAM y dos, cuatro u ocho Kbytes de ROM.

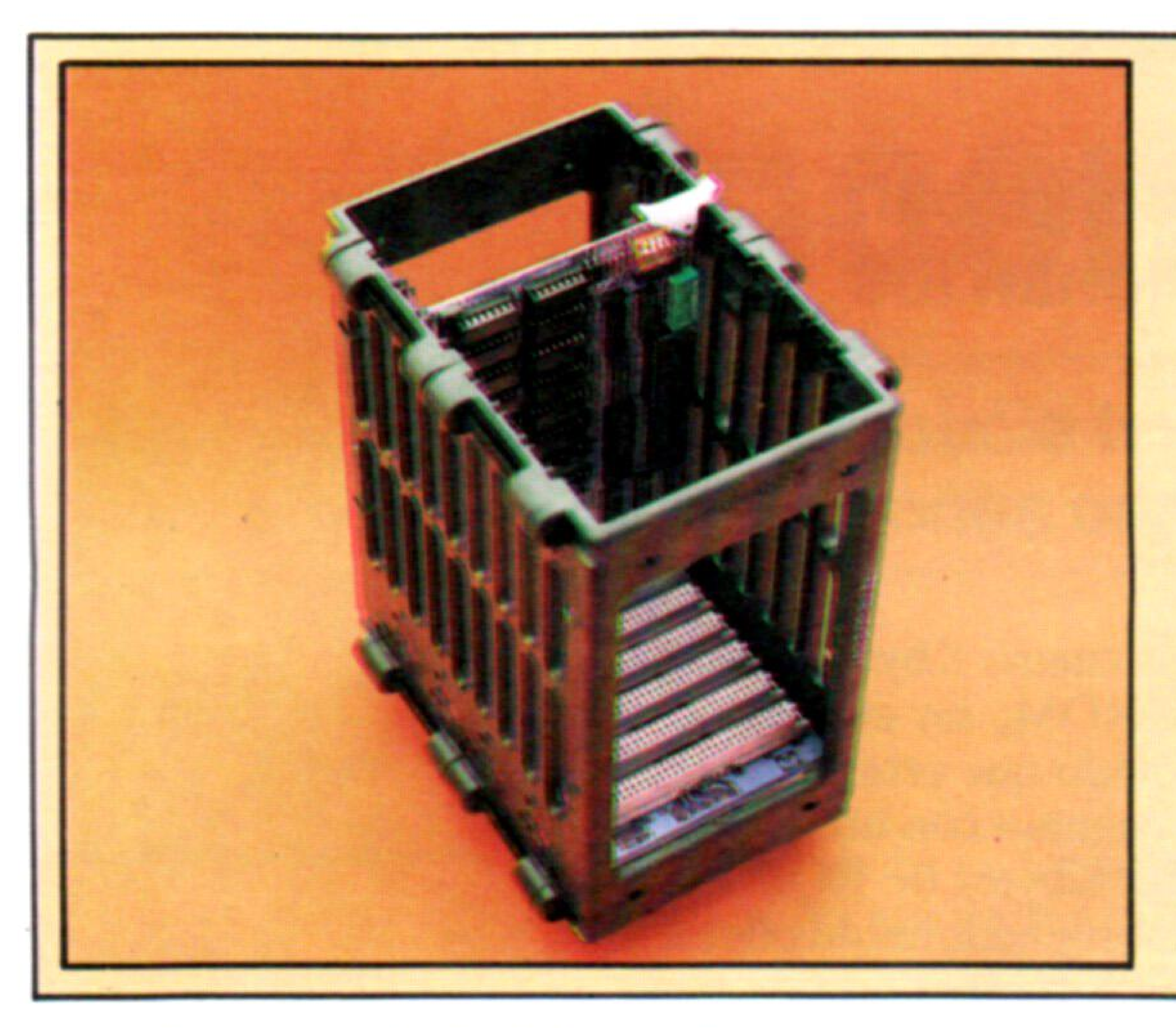
En comparación con los ordenadores de oficina y personales, la mayoría de los cuales poseen un mínimo de 16 Kbytes de RAM y a menudo mucho más, sólo cuatro Kbytes pueden parecer escasos. Pero el AIM 65 también cuenta con dos buenos y sólidos conectores de ampliación, que le permiten añadir más tableros, como, por ejemplo, el tablero de RAM estática de 32 Kbytes. En general, los sistemas experimentales se utilizan en aplicaciones que no requieren una gran capacidad de memoria y es sorprendente lo mucho que pueden hacer con los aproximadamente tres Kbytes que quedan disponibles una vez satisfechos los requisitos de operación de la máquina.

Como el AIM 65 está construido alrededor del microprocesador 6502, todas las partes del mapa de la memoria tienen límites fijos. Ello exige que los 512 bytes inferiores sean de RAM para albergar la



Chris Stevens

Veamos el AIM...
Un sistema AIM completo puede parecer algo desordenado, si bien por razones de elegancia el teclado y el tablero principal pueden presentarse en una carcasa. Es lo ideal si utiliza el AIM en su forma básica de 4 K. Para su ampliación, es posible enchufar directamente una ficha adicional en la parte posterior del AIM. Si aún es insuficiente, ha de usar un chasis de ampliación y conectarlo al ordenador mediante las dos fichas y el cable de cinta que aparecen en primer plano. La caja metálica azul es la fuente de alimentación



Tablero de memoria ampliado
Es posible ampliar el AIM casi hasta el infinito. Éste es un chasis de ampliación que puede contener todavía ocho tableros de circuitos más (también hay para cuatro y dieciséis). Son tableros de tamaño estándar y se acoplan con el conector estándar Euro de 96 patillas. Se pueden comprar fichas para ampliación de memoria (hasta 128 K de RAM y ROM), una interface IEEE estándar y fichas para poder conectar el AIM a unidades de disco flexible, monitores e impresoras de tamaño normal

zona de memoria de información transitoria, que ocupa las posiciones de bytes $0100 a $01FF. La página cero, que va del $0000 al $00FF, es una zona especial utilizada por el 6502 como "seudorregistros". Puesto que los decodificadores de dirección no necesitan incluir la dirección de página (binario 00000000) en la instrucción, en realidad ésta es más corta y mucho más rápida que las regulares. Esta característica forma parte del secreto de la potencia y la popularidad del 6502, pues da al chip 256 registros definibles por el usuario.

Como ya se ha señalado, el AIM tiene una buena cantidad de ROM en comparación con otras máquinas experimentales. Está situada en el extremo superior de la memoria, también en este caso porque es allí donde la localiza el 6502. Dispone de una selección de lenguajes y programas de utilidad, si bien todos dependen del monitor ROM, un bloque de cuatro Kbytes para utilidades de sistema de bajo nivel que reside entre $E000 y $EFFF. Cuenta con un editor de línea, espaciador paso a paso (que ejecuta el programa de a una instrucción por vez, de modo tal que es posible examinar el contenido de la memoria y, probablemente, cambiarlo en cada etapa) y localizador (que visualiza el contenido del registro de posiciones del programa a cada paso durante su ejecución), así como las características usuales de alteración de registro y memoria.

El microordenador se presenta con cinco conectores de ROM, en los que se pueden enchufar diversas opciones de firmware (programas grabados en ROM). Entre éstas existe una versión de BASIC, con números de coma flotante de cinco bytes y funciones aritméticas definidas en una línea.

Para el comprador potencial del AIM 65, que probablemente está interesado en el control y supervisión de procesos industriales u otras aplicaciones afines, resultan mucho más interesantes los otros conjuntos de ROM. Incluyen un ensamblador y un interesante pero poco conocido lenguaje denominado PL/65, que se parece un poco al ALGOL o PL/1 y se compila en un código fuente ensamblador (véase p. 596). Puede ser manipulado, con el fin de "pulirlo", y ensamblado por el ensamblador.

El AIM 65 también dispone de INSTANT PASCAL, un lenguaje poco corriente que ha sido incomprensiblemente desatendido. A diferencia de casi todas las demás versiones de PASCAL, es interactivo e interpretado y ofrece casi todas las ventajas de la estructuración así como la conveniencia y la flexibilidad del BASIC. Sin embargo, el PASCAL es un lenguaje extenso y no encaja a menos que el AIM 65 se amplíe. También dispone de FORTH, que ofrece prácticamente las mismas ventajas. Puesto que exige del ordenador menos que el PASCAL, puede utilizarse sin ampliación.

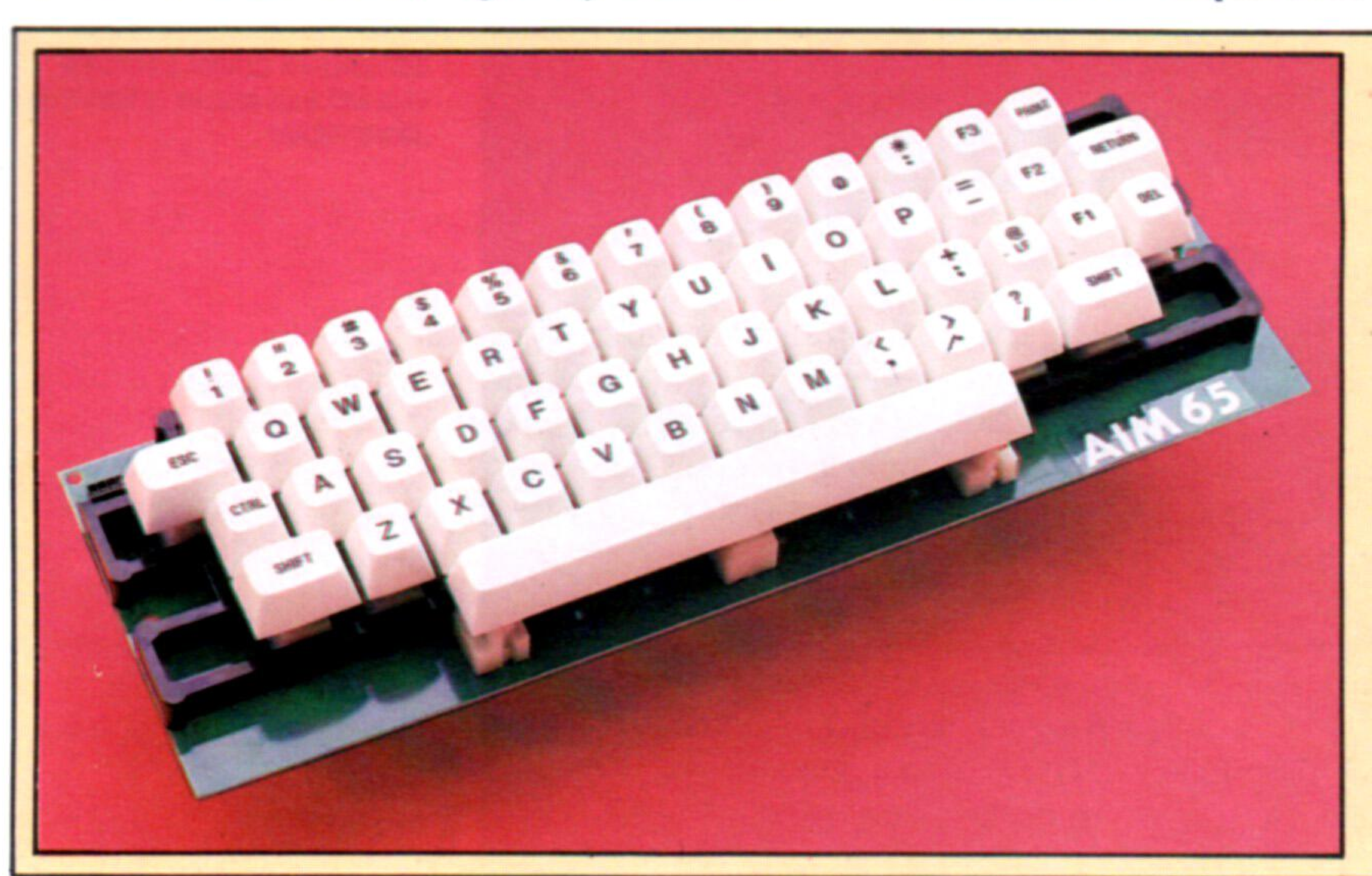
Conector de aplicación
Este conector terminal tiene dos puertas para conectar el AIM a cualquier dispositivo por controlar

Conector de electricidad
La energía eléctrica necesaria para el AIM (5 v para el ordenador y 24 v para la impresora) se conecta aquí

Impresora
El AIM se sale de los cánones usuales puesto que lleva incorporada una impresora térmica de 20 columnas. Pero resulta práctica para ciertas aplicaciones de control que exijan un permanente registro de los pasos

Interruptor de reactivo
Se emplea para volver a poner en marcha el ordenador

Decodificador del teclado
Un chip RIOT 6532 controla el teclado



Teclado
El AIM tiene un eficaz teclado que permite programar fácilmente la máquina estándar en BASIC y otros lenguajes; muchos sistemas equivalentes sólo disponen de teclados hexadecimales (sólo 0-9, A-F y unas pocas teclas más) y, por ende, no son fácilmente utilizables. La tecla PRINT del extremo superior derecho se emplea conjuntamente con la tecla CTRL para conectar y desconectar la impresora incorporada. Si la impresora está conectada, todo lo que aparece en la pantalla se imprime en la impresora

Chris Stevens

Chips VIA 6522

Conector de ampliación
Este conector terminal capacita la conexión con un chasis de ampliación o para un módulo de ampliación de memoria

RAM
El AIM tiene espacio para una RAM de hasta 4 K sobre el tablero. Puede ampliarse la memoria con módulos enchufables o con chasis de ampliación

CPU
Como la mayoría de las máquinas coetáneas al AIM, el procesador es un 6502

ROM
Se pueden acoplar hasta 20 K de ROM. Así como el monitor es estándar, el BASIC es opcional, y pueden también ajustarse un intérprete FORTH, un ensamblador o un programa completo de aplicaciones

Monitor
Programa monitor que sirve para escribir y comprobar programas en lenguaje máquina

Visualización
Esta pantalla incorporada permite el uso del AIM sin televisor o monitor

Teclado
El AIM presenta un teclado ASCII de recorrido completo, lo que significa una mejora de las teclas hexa que ostentan buena parte de máquinas similares

Interruptor Run/Step
Permite que el ordenador ejecute (Run) una instrucción al modo habitual o bien por pasos (Step) con el fin de eliminar errores

AIM 65

DIMENSIONES
292 × 267 × 60 mm

CPU
6502

MEMORIA
ROM de 4 K, con enchufes para un máximo de 20 K, y RAM de 4 K, ampliables a 64 K

PANTALLA
No hay visualización de video, sino una visualización LED de 20 caracteres y 16 segmentos. Se necesita una ficha de control para establecer una representación visual normal

INTERFACES
Dos puertas bidireccionales de ocho bits, cada una de las cuales tiene dos líneas de control, más bus de sistema

LENGUAJES DISPONIBLES
Se presenta con miniensamblador y editor de línea; también puede disponer de ensamblador completo, BASIC, FORTH, PL/65 e INSTANT PASCAL

TECLADO
53 teclas tipo máquina de escribir, con tres de función

DOCUMENTACION
Manuales de instalación y de hardware que constituyen un modelo de claridad y amplitud y contienen todos los detalles que el usuario pueda necesitar

VENTAJAS
La más importante radica en la extremada flexibilidad de la máquina, atribuible a su capacidad de ampliación y su acceso a diversos lenguajes. La documentación es superlativa

DESVENTAJAS
No presenta ninguna grave. Hay una relativa falta de software de alto nivel, pero hay que reconocer que el AIM 65 es un ordenador de investigadores más que una máquina de aplicaciones

Además del conector de ampliación, que es portador de todas las señales principales —incluida la información y las líneas de dirección, las señales horarias y las líneas de energía—, existen dos chips Versatile Interface Adapter 6522, uno de los cuales se utiliza para controlar la impresora, la interface del teletipo y la interface de la cassette. El otro no está comprometido y aparece en el conector de aplicaciones J2 como dos conexiones bidireccionales de línea ocho más dos. También se ofrece un cronómetro: el 6532 RAM I/O Timer (RIOT), pero este complejo y avanzado dispositivo se destina exclusivamente al manejo del teclado.

En tanto máquina experimental, el AIM 65 no se limita a ejecutar en BASIC o cualquier otro lenguaje de alto nivel. Se prevé que los usuarios programarán sus propias ROM especializadas, que luego podrán conectar en los enchufes de ROM, dedicando así la máquina a un trabajo especializado.

En líneas generales, el AIM 65 es un ordenador de tablero único robusto, flexible y bien sustentado y diseñado. Cuenta con medios suficientes para llamar la atención de todo el que necesite una máquina pequeña, pero muy útil, sin las características de un ordenador de oficina estándar, pero con mayor flexibilidad de la propia de un ordenador personal.

Departamento de contabilidad

Esta nueva serie analizará el mercado de paquetes de gestión, abierto tanto a pequeños usuarios del comercio como a empresas y sociedades

La calificación "paquetes de gestión" abarca un amplio campo de aplicaciones y, además de la diversidad en los servicios, se ofrecen en múltiples proporciones. La primera diferencia en proporción importante es la existente entre el software de cassette dirigido al usuario de ordenadores personales y el software de disco. Como la tecnología de almacenamiento en disco da a los programadores acceso a medios de lectura y escritura rápida, el software de gestión debería basarse en discos, pero no todo el mundo está dispuesto a pagar precios adicionales bastante altos para adquirir un sistema de disco de libre rotación.

Otra distinción es la que existe entre microordenadores de un solo usuario y máquinas de usuarios múltiples, es decir, los microordenadores que pueden sustentar cierta cantidad de terminales separados. Comenzaremos por los sistemas más sencillos y más adelante abordaremos las aplicaciones de las máquinas de usuarios múltiples.

El propósito del software de gestión es que sea utilizado por el profano más que por el especialista en informática. En virtud de ello, ya se trate de un programa diseñado para el sistema de cassette o para el más amplio sistema de disco de usuarios múltiples, por lo general adoptará un enfoque "interactivo". Esto significa que a lo largo de todas las etapas de operación del sistema el usuario es guiado por una serie de orientaciones y mensajes que aparecen en la pantalla. Por lo común, las diversas operaciones se dispondrán como un conjunto de opciones numeradas en forma de "menú". Seleccionar una función del menú principal podría llevar al usuario a un menú secundario o incluso terciario de elecciones. A continuación la pantalla visualizará una solicitud de datos, generalmente información parecida a la que se maneja en un sistema manual de contabilidad. Este uso "sin esfuerzo" es uno de los factores del enorme éxito que los programas aplicados a la gestión están teniendo.

Las tres divisiones funcionales que abordaremos en esta serie son el libro de ventas, el libro de compras y el libro mayor. El libro de ventas consigna los ingresos por ventas realizadas durante el ejercicio económico. El libro de compras registra los gastos por compras necesarias para el tráfico comercial. El libro mayor ofrece una visión global del estado del negocio y su situación financiera en cualquier momento dado.

Los programas de gestión se componen de archivos. Cada archivo contiene cierto número de registros, y cada registro cierto número de campos. La distinción entre archivos, registros y campos es notable. Por ejemplo, un programa de libro de ventas tendrá un archivo maestro de clientes que se compone de los registros de cada uno de los clientes. Los detalles sobre cada cliente, tales como nombre y señas, se dividen en campos dentro del registro.

Además, un programa contable debe seguir un modo preciso en el tratamiento de los datos. Ejemplos de las rutinas de programación necesarias son las rutinas de entrada (que permiten introducir los datos en el sistema) y las rutinas aritméticas, que manipulan valores numéricos en campos numéricos dentro y a través de los registros.

En los "macropaquetes" de gestión (diseñados para ser utilizados con sistemas de discos) se regis-

Administración del dinero
Un remedio, aceptable ante las limitaciones de las cassettes, sería la venta de programas independientes que, en un momento dado, puedan aglutinar la información. Este programa de facturación funciona por sí mismo, dejándole a usted la tarea de actualizar manualmente los registros de existencias. Si lo prefiere, puede utilizar archivos generados por los programas adjuntos como parte de un sistema totalmente automatizado. Una ventaja de este enfoque reside en que gradualmente puede ir informatizando su empresa en lugar de someterla a un brusco cambio de sistema

Libro mayor simplificado
El software de gestión en cassette es limitado en cuanto a la magnitud y número de programas contenidos en un paquete. En consecuencia, algunos programas como éste suelen concentrarse en una tarea específica. El paquete contiene programas de contabilidad especiales para cuentas corrientes, resúmenes anuales, etc. Pero sólo se propone ser una ayuda administrativa para controlar el movimiento del dinero y no manejará por sí mismo todas sus cuentas

Liz Heaney

tran gran cantidad de detalles en los libros de ventas y compras. Así, cada venta o compra se registra en la cuenta de cada cliente o de cada proveedor.

Incluso el más grande sistema para microordenador tiene un límite máximo en cuanto a la cantidad de datos que puede almacenar, dado por descontado que los paquetes de cassette no pueden hacer frente a los detalles de tantas transacciones como los paquetes de disco. En consecuencia, una de las preguntas clave que el usuario potencial de este tipo de software en cassette (y también de software en disco) tiene que hacerse es: ¿podrá este sistema hacer frente al volumen de trabajo relacionado con mi negocio?

Cuantos más datos se almacenan en un archivo, más tarda el ordenador en seleccionarlos, por lo que la solución más común consiste en que los programas sólo retengan información detallada de las transacciones pendientes de pago. Los sistemas basados en este principio se denominan libros mayores *abiertos*.

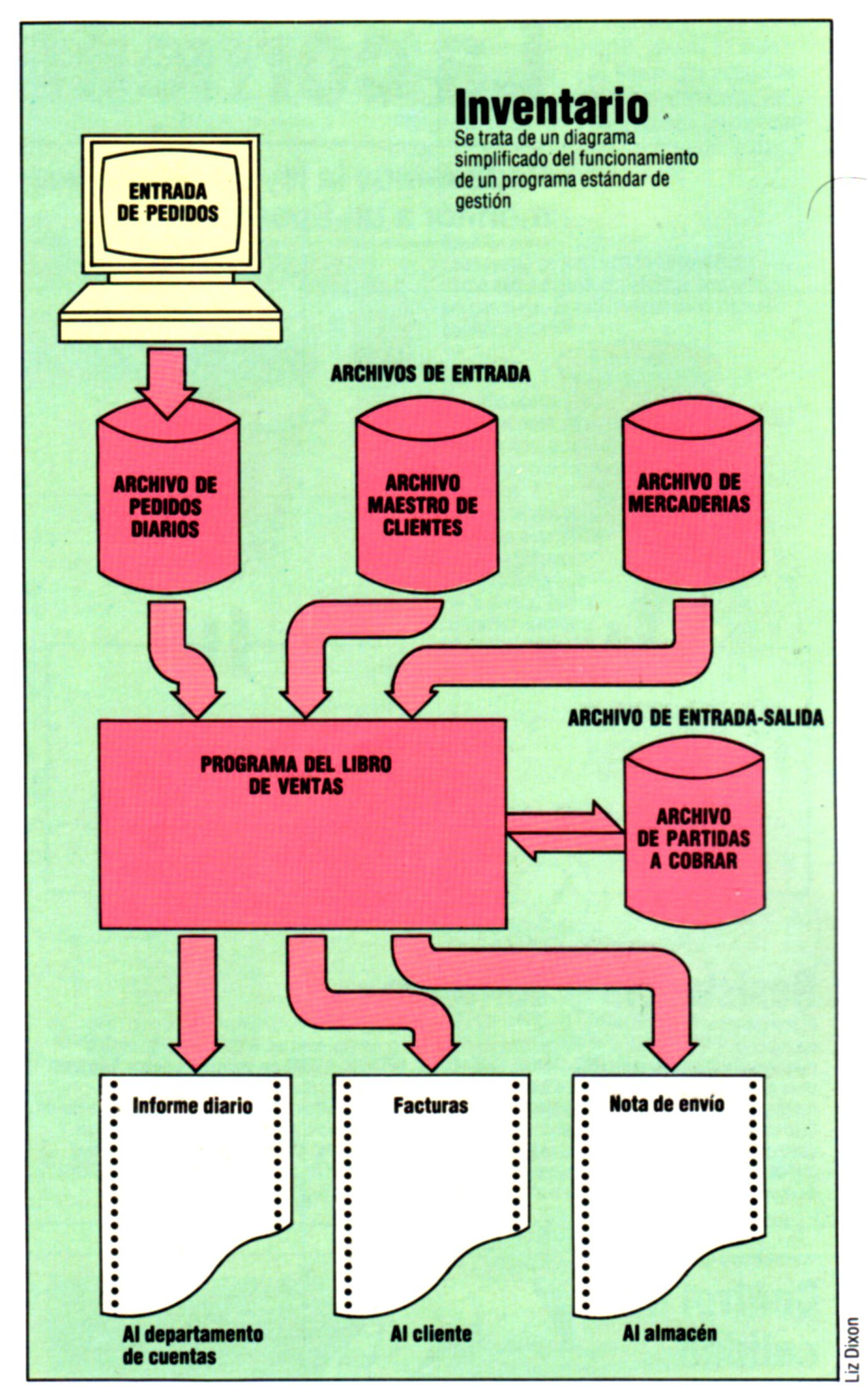
Otra solución suele ser que el ordenador sólo retenga el valor acumulativo de las transacciones entre la empresa y sus clientes o proveedores. Se denomina contabilidad por balances de saldos. Aunque transmite menos información, necesita menos memoria y es más fácil de operar. En resumen, existe un compromiso entre eficiencia (respecto al nivel de detalle que puede registrarse) y lo que se desearía en cuanto a capacidad de procesamiento y memoria de la máquina.

Como veremos a lo largo de las próximas páginas de esta serie, los programas de disco para libros de ventas y compras poseen rasgos específicos que no es posible incorporar en los paquetes de cassette. Por ejemplo, los programas del libro de ventas pueden incluir hasta una opción de facturación que permite a la empresa generar facturas y estados de cuenta para enviarlos a los clientes. Los programas del libro de compras pueden incluso imprimir cheques y acuses de recibo (haciendo una lista de lo que ha entrado) para remitir a los proveedores.

El software de gestión cumple una serie de funciones. El modo más sencillo de comprenderlas consiste en considerar la forma en que el dinero entra y sale de una empresa. El requisito básico de todo el que haga un negocio consiste en averiguar cuántos ingresos tiene la empresa y cuántos gastos ocasiona. El método más sencillo para saberlo es el libro de caja. Un libro de caja no informatizado sino manual lleva las páginas divididas en tantas columnas como necesita la empresa para identificar los conceptos por los que paga o cobra dinero. El libro de caja también tendrá que consignar el valor del IGTE, o del IVA (impuesto sobre el valor añadido) a efectos fiscales.

El libro de caja suele llevarse diaria o semanalmente. Se pueden apuntar las transacciones una a una o, como es más corriente, hacer el apunte del valor total de la recaudación diaria. La diferencia entre el simple libro de caja y un sistema de cuentas completo centrado en el libro mayor y ayudado por libros de ventas, de compras y gastos generales, radica en la falta de detalles del primero.

Los programas de cassette especiales para teneduría de libros disponen de una cantidad muy reducida de memoria. No tienen un disco donde escribir los datos para despejar a continuación una zona de espacio de trabajo de la memoria interna que acepte nuevos datos. Por esto los programas de cassette suelen adoptar el sistema resumido del libro de caja.



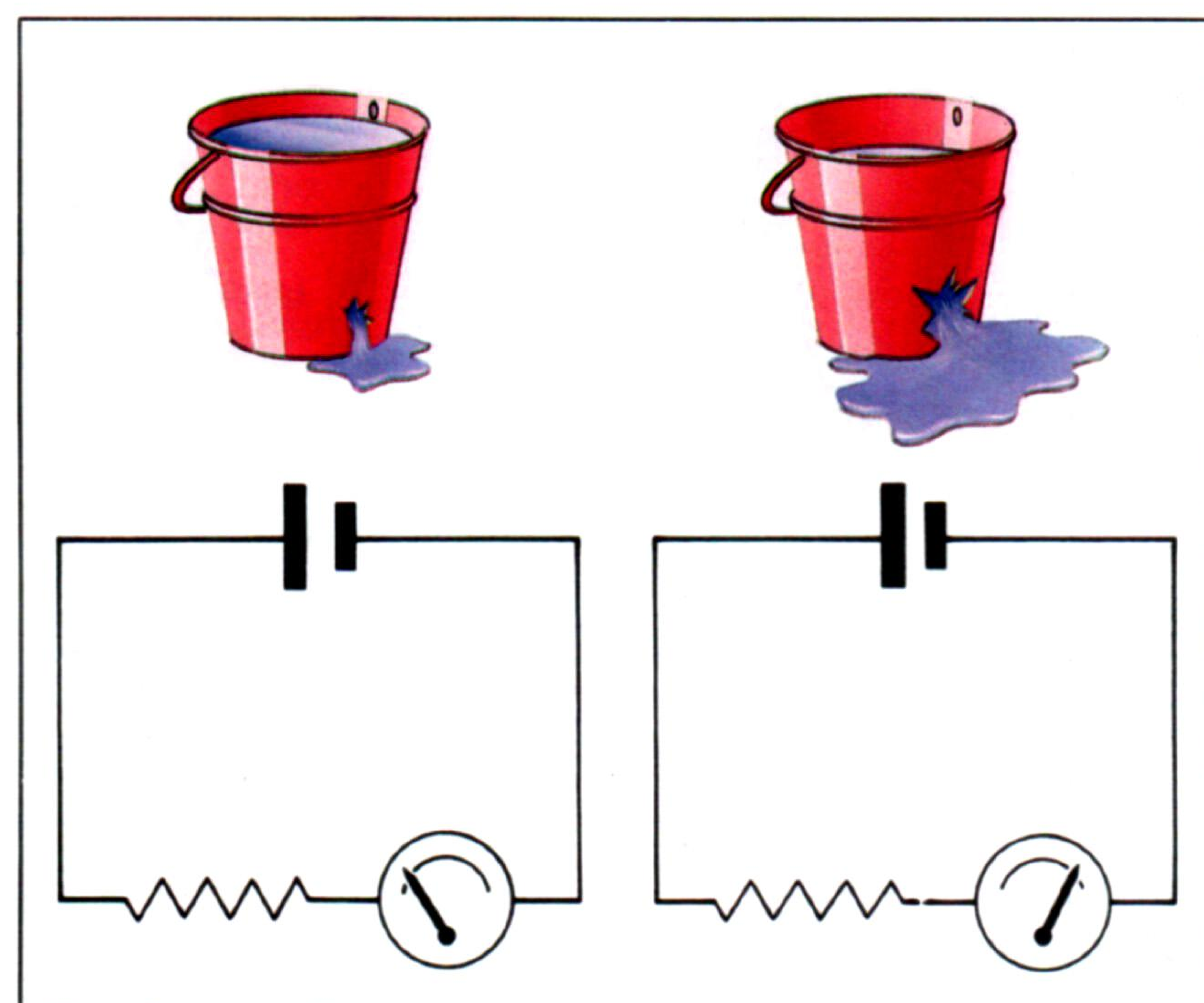
Inventario Se trata de un diagrama simplificado del funcionamiento de un programa estándar de gestión

En lugar de contar con tres programas de cassette por separado correspondientes a los libros de ventas. de compras y general, dichos programas suelen unir los tres. Un ejemplo típico sería el programa que lleva cuenta de los ingresos y gastos de un pequeño comerciante en apuntes semanales. Tal sistema tendría como objetivo final unificar y sintetizar los datos incorporados semana tras semana. Para ello tiene que sumar todo el dinero cobrado y pagado durante la semana, restar el segundo del primero y presentar un informe sobre el resultado, favorable o desfavorable, a caja.

En el próximo capítulo analizaremos los requisitos de diseño de dicho sistema.

La corriente y sus usos

Repasemos la ley de Ohm y veamos cómo se puede conectar un monitor a un Spectrum

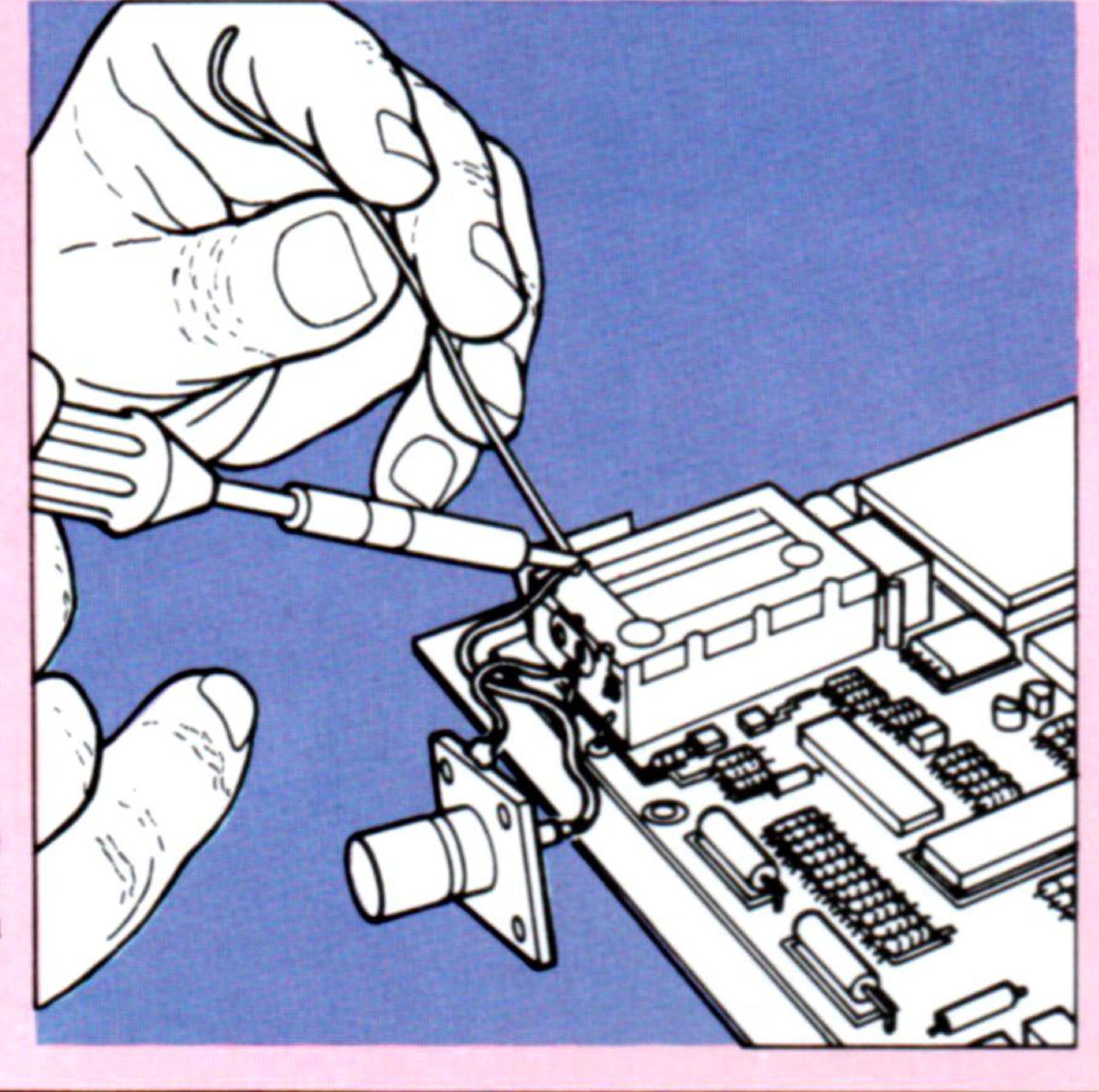


Resistencia y corriente

El componente de un circuito eléctrico ofrece resistencia al flujo de electricidad, del mismo modo que las tuberías y los depósitos de un sistema de calefacción central resisten el flujo de agua: cuanto mayor sea la resistencia ofrecida, menor será el flujo. Si un cubo tiene un pequeño agujero, escapará poca agua debido a que los bordes del agujero resisten el flujo; si el agujero es más grande, la resistencia se reduce y fluye más agua. Así, un circuito eléctrico que porta una alta resistencia tiene un flujo de corriente menor que el que tendría con una baja resistencia. La resistencia eléctrica de un componente depende en parte del material con que está fabricado: los cables son de cobre porque ofrece poca resistencia; los filamentos de las bombillas se hacen de un material altamente resistente, de modo que al pasar la corriente por la resistencia se genere calor y, en consecuencia, luz

Control de calidad

Mediante esta sencilla modificación, el Spectrum de Sinclair proporciona una señal compuesta de video, que permite usar monitor de alta resolución. Abra la caja e identifique el modulador RF, un voluminoso componente plateado situado en el extremo superior izquierdo. El modulador presenta dos conexiones. Busque la más cercana a la parte posterior. Suelde un trozo de hilo conductor desde ésta hasta la parte exterior de un enchufe de soporte de superficie BNC. Suelde otro conductor desde la caja del modulador hasta el conector central del enchufe. Monte éste a un lado de la caja



ATENCIÓN: La garantía de su ordenador quedará anulada si abre la caja

En el último capítulo de la serie *Bricolaje* repasamos algunos de los fenómenos que hacen funcionar un sistema eléctrico. A partir de este momento profundizaremos en la relación existente entre los tres componentes teóricos de la electricidad: potencial, corriente y resistencia. Para ello estudiaremos el modo en que se relacionan y la sencillísima operación aritmética que rige tal relación.

El físico alemán Georg Simon Ohm (1789-1854), que en 1827 enunció la ley fundamental de las corrientes eléctricas que lleva su nombre, fue el primero en deducir la relación entre potencial, corriente y resistencia, y en comprender que cualquiera de los tres valores fundamentales puede deducirse a partir de los otros dos. Comprendió que el potencial era directamente proporcional a la resistencia y a la intensidad de corriente, y una simple operación aritmética le permitió modificar la ecuación, de modo que era posible determinar la resistencia dividiendo el potencial por la intensidad de corriente, y calcular la intensidad de corriente dividiendo el potencial por la resistencia.

Mediante el empleo de los símbolos convencionales de una sola letra V por potencial, I por intensidad de corriente y R por resistencia, las ecuaciones propiamente dichas resultan muy sencillas, y se expresan como sigue:

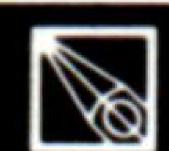
$$V = R \times I$$
$$R = V/I$$
$$I = V/R$$

Una extensión de la ley de Ohm es la ley de potencia:

$$W = V \times I$$

que relaciona W, la potencia consumida en un circuito o dispositivo, con el potencial y la intensidad de la corriente que lo recorre. Si el potencial en esta fórmula se expresa en voltios y la corriente en amperios, W (la potencia) se mide en vatios.

Esta ecuación es muy útil en el hogar pues nos permite determinar el valor del fusible que debemos aplicar a un electrodoméstico. Tomemos como ejemplo un conector de tres kilovatios, que funciona a 220 voltios. Si dividimos 3 000 (la potencia consumida por el electrodoméstico, expresada en vatios) por 220 (la tensión de la red eléctrica), tenemos que la cantidad de corriente consumida será de 13,6 amperios. Por lo tanto, cualquier fusible de valor inferior a 13,64 amperios saltará. Puesto que el valor máximo de cualquier enchufe doméstico es de 13 amperios, también podemos deducir que cualquier otro dispositivo, como un calefactor de aire de tres kilovatios, que funcione con un adaptador conectado al mismo enchufe no debe consumir más de 120 vatios o estaremos sobrecargando el circuito. En el próximo capítulo llevaremos a cabo algunos experimentos a fin de demostrar prácticamente la ley de Ohm.



Piezas de un circuito

El diagrama de un circuito electrónico parece complicado para quien nunca lo ha visto. A fin de ilustrarlo, hemos elegido el circuito simple de un intercomunicador y descrito uno por uno sus componentes electrónicos. Cada uno de ellos está representado por un símbolo determinado. Cada componente del diagrama también se identifica mediante un código, como por ejemplo "R1" o "TR2". Es un modo práctico de referirse a los componentes, por ejemplo, en una lista de piezas de recambio. Las líneas que conectan los componentes representan alambres. Se dibujan en línea recta por razones de claridad y podrían ser verdaderos alambres o líneas de estaño de un circuito impreso

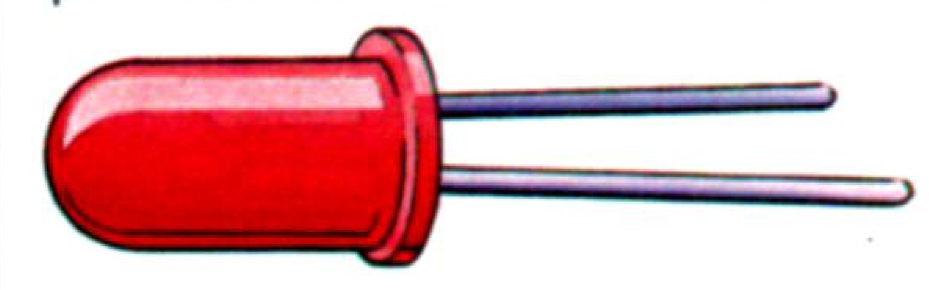


1 Interruptores
Existen múltiples variedades de interruptores, cada uno con su propia función. Los dos tipos principales son los de retención, donde el estado del interruptor se mantiene al ser presionado, y los de no retención, donde el contacto sólo tiene lugar mientras se mantenga la presión

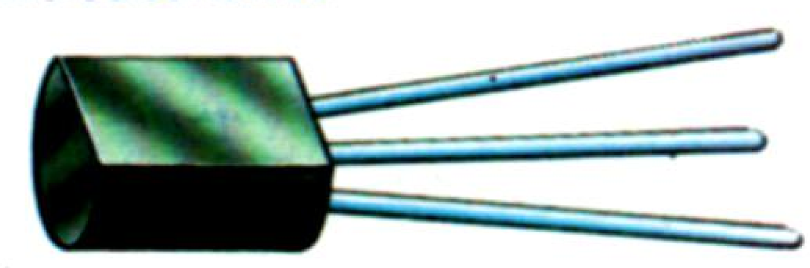
2 Diodos emisores de luz
Los diodos constituyen la forma más simple del semiconductor. Son el equivalente electrónico de una válvula sin retorno y sólo permiten que la corriente fluya en una dirección. Algunos diodos encapsulados en una resina translúcida despiden una pequeña cantidad de luz y por ello son útiles como indicadores

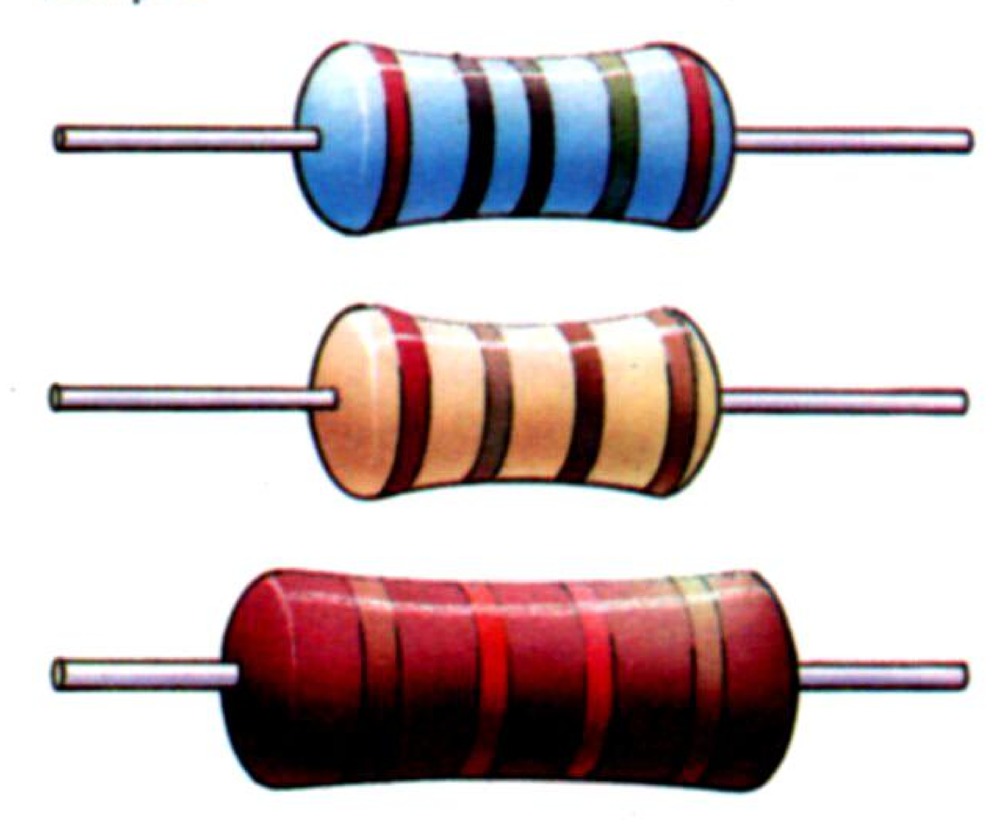


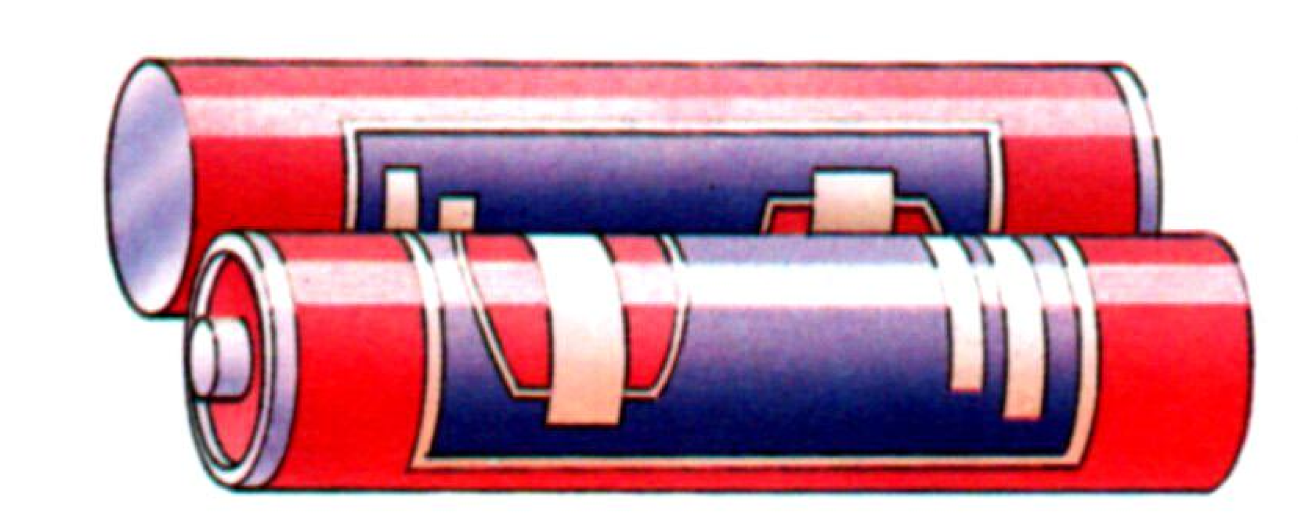
3 Transistores
Según su especificación y la forma en que se emplea, el transistor puede hacer las veces de interruptor o de amplificador. Su invención, en 1947, allanó el terreno para posteriores desarrollos en el campo de la microelectrónica

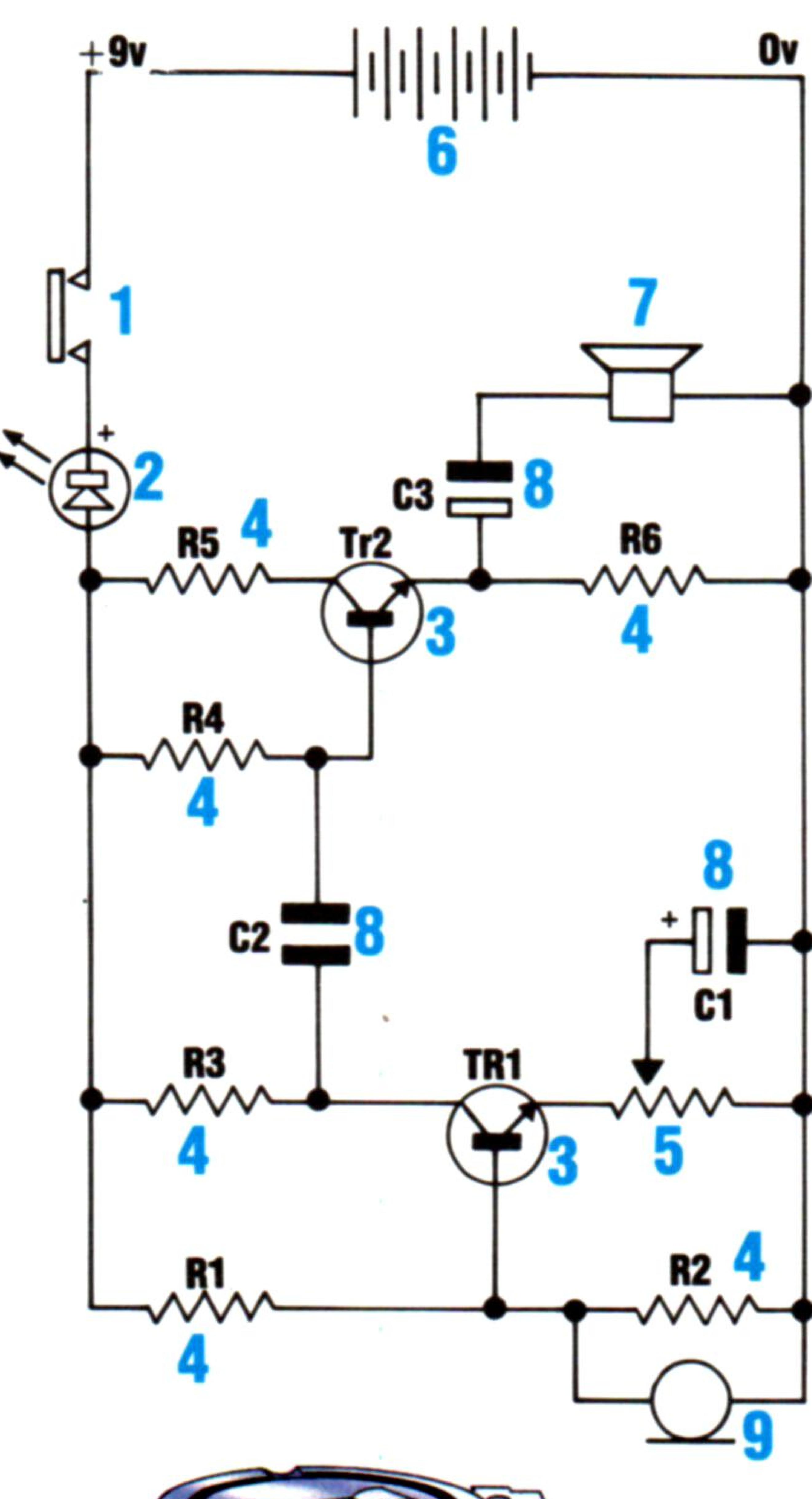


4 Resistencias
Si introducimos en el circuito materiales menos conductores, podemos utilizarlos para controlar el flujo de electricidad. Las resistencias se presentan en muchos tamaños distintos. Su valor se expresa mediante listas de color alrededor de sus cuerpos





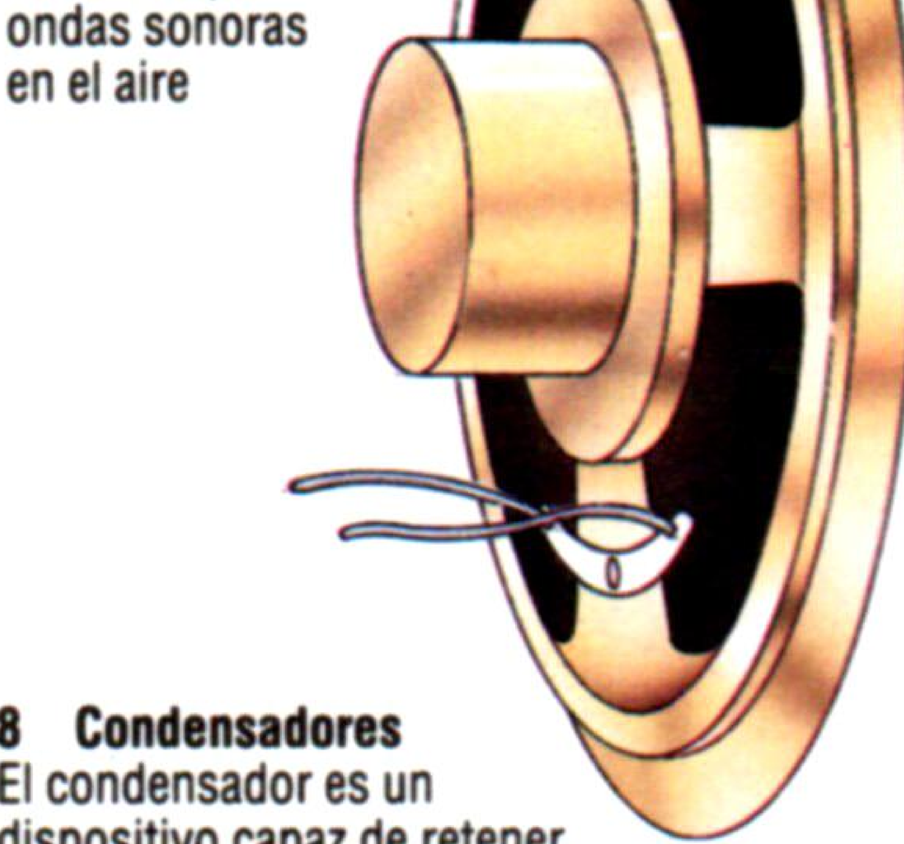




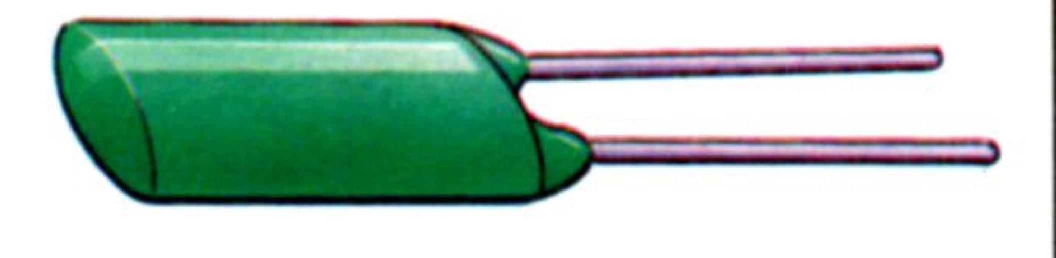
5 Resistencias variables
No todas las resistencias son constantes. Las resistencias variables, en ocasiones denominadas potenciómetros, utilizan una faja de carbono como conductor. La distancia que la corriente ha de recorrer a través del carbono antes de llegar al terminal de despegue determina la resistencia del componente

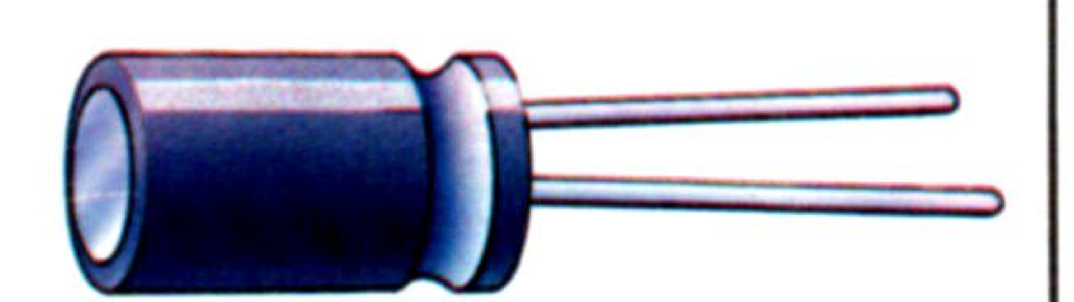
6 Pilas
La mayoría de los circuitos pequeños como éste pueden alimentarse con pilas corrientes pues suministran una corriente continua estable

7 Altavoces
Los altavoces son parientes directos de los micrófonos: operan del mismo modo pero obtienen resultados opuestos. En un altavoz, el diafragma adosado a un imán vibra en respuesta a la corriente que se le aplica. Dicha vibración produce ondas sonoras en el aire

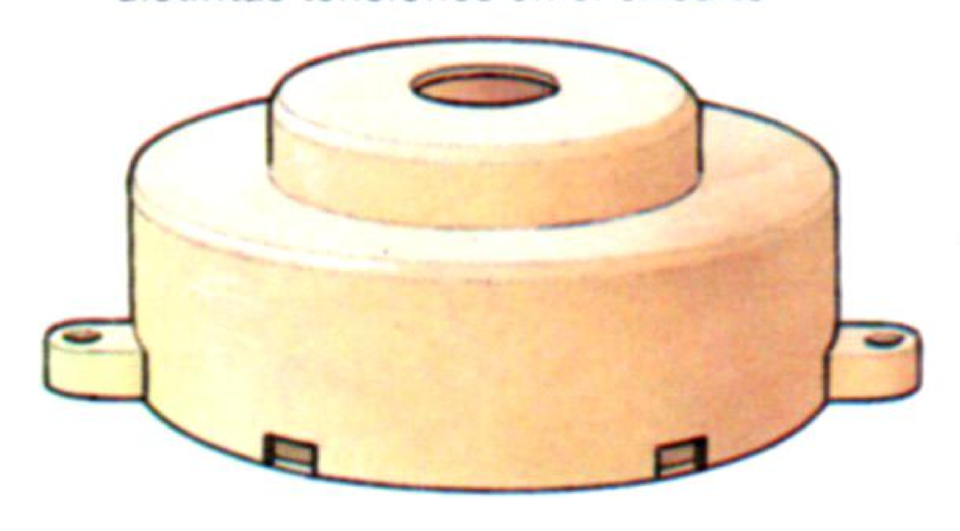


8 Condensadores
El condensador es un dispositivo capaz de retener una carga eléctrica. El condensador se carga cuando sus dos terminales están conectadas a una fuente de alimentación. Una vez plenamente cargado, no ocurre nada, ni siquiera cuando cesa la alimentación, hasta que ambos terminales del condensador se conectan, momento en que se descarga





9 Micrófonos
Los micrófonos funcionan de manera inversa a los altavoces. Las ondas sonoras hacen que el diafragma vibre y esta vibración, a través del imán, produce distintas tensiones en el circuito



Kevin Jones

De byte en byte

Analizaremos el desarrollo de un programa, desde la definición de la tarea inicial hasta el lenguaje máquina propiamente dicho

En los capítulos precedentes hemos visto de qué manera a la entrada de las líneas de un programa BASIC éstas se reducen a distintivos (*tokens*) seguidos de datos ASCII. A partir de ello comprendimos que el BASIC, aunque ciertamente es un lenguaje de alto nivel, no es tan elevado: se compone, básicamente, de secuencias de instrucciones y cada instrucción consiste en una orden (inmediatamente reemplazada por un distintivo, que en sí mismo sólo está un escalón por encima de un código operativo máquina) seguida de los datos para dicha orden. El hecho de que las órdenes y sus datos (variables, números o cadenas) estén más próximos al lenguaje natural y de que las instrucciones se encuentren visiblemente separadas por números de línea o por dos puntos, hace que a nosotros un programa en BASIC nos parezca de mucho más alto nivel que el intérprete de BASIC. De ello se deduce que el lenguaje máquina sólo necesitará unos pequeños "retoques" para que resulte razonablemente comprensible a nuestros ojos.

Estos "retoques" del lenguaje máquina son los creadores del lenguaje assembler, en el que un alfabeto mnemotécnico como LDA y ADC representan los códigos operativos (opcodes) de byte único que el microprocesador comprende realmente, y en el que pueden emplearse símbolos alfanuméricos como LABEL 1 y TFLAG en lugar de direcciones de memoria y datos numéricos. Puesto que el microprocesador no comprende el lenguaje ensamblador, antes de poder ejecutar un programa hay que traducirlo a lenguaje máquina, ya sea mediante un programa denominado ensamblador o manualmente por parte del programador. La utilidad del lenguaje ensamblador consiste en que es lenguaje máquina traducido. Mediante la simple sustitución de códigos operativos por mnemotecnia y números por símbolos, lo podemos convertir directamente en un código ejecutable. Pero a nuestros ojos resulta mucho más comprensible que el lenguaje máquina y, por ende, resulta muy útil en la elaboración de programas. Siempre escribiremos los programas en lenguaje ensamblador y prácticamente no nos ocuparemos del equivalente en el lenguaje máquina hasta las últimas etapas del desarrollo de un programa. Pero en este momento vale la pena hacer ambos por su interés y por razones de total claridad, recordando que, en general, el lenguaje ensamblador hará todo lo que queramos.

El microprocesador puede llevar a cabo muchas operaciones distintas pero, fundamentalmente, lo único que hace es manipular el contenido de la memoria. Lo hace actuando directamente en la memoria del ordenador —los chips de RAM y ROM de que consta el sistema del ordenador— u operando a través de su propia memoria interna, que se compone de *registros*. Estos últimos son varios bytes de memoria físicamente localizados dentro del chip del microprocesador y que tienen determinadas funciones específicas pero que, por lo demás, no presentan ningún carácter que los distinga de los restantes bytes de memoria.

Registro acumulador

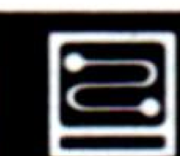
El más importante de los registros del microprocesador se denomina *acumulador*. Está directamente conectado a la ALU (*Arithmetic and Logic Unit*: unidad aritmético lógica) y se utiliza con más frecuencia que cualquiera de los restantes registros. A fin de poder emplearlo, debemos estar en condiciones de introducirle información, proceso que se denomina "cargar el acumulador" (*Loading the Accumulator*). Mediante el uso de lenguaje ensamblador, decimos que el 6502 lo hace llevando a cabo la operación LDA, y en el Z80 mediante la operación de LD A. Tomar información *desde* el acumulador es tan decisivo como cargarlo y en el lenguaje ensamblador 6502 ello se logra mediante la operación STA (*STore the Accumulator contents*: almacenar el contenido del acumulador). Empero, el Z80 considera la carga y el almacenamiento como casos distintos de lo mismo, es decir, transferencia de datos. Por lo tanto, tomar información desde el registro acumulador es algo que también se hace mediante la operación LD A, pero en un formato distinto, como veremos más adelante.

Supongamos que queremos escribir un programa de lenguaje ensamblador que copie el contenido de un byte de memoria en el siguiente. Empecemos por copiar el byte $09FF en el byte $0A00. Inmediatamente podemos expresarlo del siguiente modo:

6502	**Z80**
LDA $09FF	LD A,($09FF)
STA $0A00	LD ($0A00),A

Nótese que estamos copiando el contenido del byte $09FF en el byte $0A00 sin saber cuál es dicho contenido: es vital tener clara esta cuestión desde el principio. El byte $09FF puede contener cualquier número desde el $00 al $FF, y todo lo que nuestro programa hace es cargar en el acumulador dicho número y a continuación transferirlo desde el acumulador al byte $0A00. La versión 6502 del lenguaje ensamblador no pone de relieve que LDA se refiera al *contenido* de $09FF, pero distingue inequívocamente entre cargar (LDA) y almacenar (STA). La versión Z80 no hace esta última distinción en sus códigos operativos, pero su formato de instrucción siempre es:

```
OPCODE DESTINO (FUENTE)
```



Esta versión pone entre paréntesis direcciones de memoria cuando significa "el contenido de", lo cual refuerza la distinción esencial entre la dirección de un byte y lo que contiene.

El programa que hemos dado es lógicamente completo, pero hemos de ejecutarlo como una subrutina, de modo que para completarlo se necesita el equivalente de la orden RETURN (volver al programa desde la subrutina). Los códigos operativos (opcodes) son RTS en el 6502 y RET en el Z80.

Para usar esta subrutina, primero tenemos que traducirla a lenguaje máquina, luego almacenar el código en algún lugar de la memoria y hacer que la ejecute el microprocesador. Podemos utilizar el programa monitor (véase p. 598) para las dos últimas tareas, pero primero tenemos que realizar la traducción y decidir a dónde irá el código. Esto último es nuevo para los programadores de BASIC, que nunca han de pensar dónde se acumulará un programa de BASIC, ya que se limitan a escribirlo y a teclear RUN. Las decisiones de almacenamiento son tomadas por los diseñadores del sistema en nombre del programador y ejecutadas luego por el sistema operativo.

Un programa en lenguaje máquina puede almacenarse y ejecutarse en cualquier lugar de la memoria, si bien algunos sitios son más idóneos que otros. Los lugares seguros varían de máquina en máquina, de ahí las versiones distintas del programa siguiente:

6502		
Dirección de localización	**Lenguaje máquina**	**Lenguaje ensamblador**
COMMODORE 64		
$0350	AD 56 03	LDA $0356
$0353	8D 57 03	STA $0357
$0356	60	RTS
MICRO BBC		
$0070	AD 76 00	LDA $0076
$0073	8D 77 00	STA $0077
$0076	60	RTS
Z80		
SPECTRUM DE 16 K		
$7FA0	3A A6 7F	LD A,($7FA6)
$7FA3	32 A7 7F	LD ($7FA7),A
$7FA6	C9	RET
SPECTRUM DE 48 K		
$FFA0	3A A6 FF	LD A,($FFA6)
$FFA3	32 A7 FF	LD ($FFA7),A
$FFA6	C9	RET

Nótese que cada versión del programa copia su último byte en el byte siguiente. Por ejemplo, el programa del Spectrum de 48 K copia el contenido de $FFA6 en $FFA7. Nótese asimismo que en lenguaje ensamblador las direcciones aparecen en *hi-lo* (alto-bajo), para nuestro beneficio, pero que en la traducción a lenguaje máquina aparecen en *lo-hi* (bajo-alto). Convendría reparar especialmente en que para el Z80 la mnemotecnia es LD tanto en la primera como en la segunda instrucción, aunque los opcodes difieran: 3A para la transferencia de datos al acumulador y 32 para la transferencia desde el acumulador.

Uso del programa monitor

INICIO
Anote dirección de localización y siete bytes de lenguaje máquina
RUN MONITOR
SELECCIONE DISPLAY
Inspeccione el contenido de los bytes de la dirección de localización
SELECCIONE ALTER
Introduzca el lenguaje máquina en los bytes de localización
SELECCIONE DISPLAY
Inspeccione el contenido de los bytes de la dirección de localización
¿Código bien introducido? (NO: volver a "Inspeccione el contenido..."; SÍ: continuar)
SELECCIONE EXECUTE
Introduzca la dirección de localización
SELECCIONE DISPLAY
Compare el contenido del último byte de programa con el del byte siguiente
¿SON IGUALES? (NO: volver a "Anote dirección..."; SÍ: continuar)
CORRECTO
FIN

El programa (véase página siguiente) le permite **ALTER**, (alterar) **DISPLAY** (visualizar) y **EXECUTE** (ejecutar) memoria. Cada vez que seleccione una de estas opciones, se le pedirá una dirección hexadecimal. Es la dirección de la memoria donde:

1) Se comienza a alterar el contenido de la memoria, o

2) Se comienza a visualizar el contenido de la memoria, o

3) Se comienza a hacer que el microprocesador ejecute un programa en lenguaje máquina.

Al introducir "X" en lugar de un número o dirección siempre se le remitirá al nivel de mando y en éste "Q" pondrá fin a la ejecución del programa.

Si selecciona el modo **ALTER** y ha dado la dirección cuyo contenido desea alterar, dicha dirección se visualizará seguida de un interrogante; escriba el nuevo contenido hexadecimal y teclee **RETURN**. La dirección del byte siguiente se visualizará de modo semejante para ser alterada. Mientras quiera alterar bytes, siga escribiendo el nuevo contenido y tecleando **RETURN**. Si incorpora la letra X en lugar de una dirección, será remitido el nivel de mando.

Cargar en la memoria
Observe el programa para su máquina, anote la primera dirección de localización y los siete bytes en lenguaje máquina (p. ej., $0350 y AD, 56, 03, 8D, 57, 03, 60 para el Commodore 64): utilizará el programa monitor para cargar estos siete números hexadecimales en los siete bytes de memoria a partir de la última dirección de localización.

1) RUN el programa monitor, **DISPLAY** el contenido de la memoria desde la primera dirección de localización en adelante (p. ej., del $0350 al $0357 en el caso del Commodore 64).

2) Seleccione **ALTER**, dé entrada a la dirección de localización y los siete bytes de que consta el programa en lenguaje máquina.

3) Vuelva a seleccionar **DISPLAY** y cerciórese de que ha ingresado correctamente los bytes en lenguaje máquina en los bytes de dirección de localización.

4) Seleccione **EXECUTE** e introduzca la dirección de localización: parecerá que no ocurre nada.

5) Seleccione **DISPLAY**, inspeccione las direcciones de localización y verá que el contenido del último byte del programa ha sido copiado en el byte siguiente



Programa monitor para el Spectrum

```
48 REM ************************************
49 REM*                                   *
50 REM*        HCAC MONITOR 1             *
51 REM*       ——SPECTRUM——                *
52 REM*   GUARDE ESTE PROGRAMA            *
53 REM*    ANTES DE EJECUTARLO            *
54 REM*                                   *
55 REM ************************************
100 GOSUB 1000 :REM *INIC*
200 CLS
300 PRINT "**HCAC MONITOR 1 CO
    MMANDS **"
400 FOR P = 1 TO LT:PRINT O$(P):NE
    XT P
500 FOR Z = 0 TO 1 STEP 0
550 GOSUB 2000 :REM *INPUT*
600 GOSUB (4500 + CM*500)
650 NEXT Z
700 STOP
750 REM****FIN PROG PRINC****
1000 REM*****S/R INIC******
1050 LET LT = 4:DIM C$(LT):DIM O$(
     LT,24):DIM X$(16)
1100 LET X$ = "0123456789ABCDEF"
1150 LET C$ = "ADGQ":LET C1 = -48:LE
     T C2 = 10-CODE (C$(1))
1200 LET O$(1) = " A - ALTER
     MEMORY"
1220 LET O$(2) = " D - DISPLA
     Y MEMORY"
1240 LET O$(3) = " G - EXECUT
     E M/CODE"
1260 LET O$(4) = "                Q - EXIT
     PROGRAM"
1300 RETURN
2000 REM*****S/R INPUT******
2100 FOR P = 0 TO 1 STEP 0
2150 PRINT:PRINT"COMMAND ??"
2190 IF INKEY$<>" " THEN GO TO 21
     90
2200 LET A$ = INKEY$:IF A$ = " " THEN
     GO TO 2200
2250 FOR J = 1 TO LT
2300 IF A$ = C$(J) THEN LET CM = J:L
     ET J = LT:LET P = 2
2350 NEXT J:NEXT P:IF A$ = "Q" THE
     N RETURN
2400 PRINT O$(CM)
2450 FOR P = 0 TO 1 STEP 0
2500 INPUT "DIRECCION HEXA (X = FUERA)"
     ;A$
2550 GOSUB 5200 :REM CHK&ADJ
2600 NEXT P:IF A$ = "X" THEN LET C
     M = 0
2650 RETURN
3000 REM****S/R BYTE HEXA****
3010 LET HB = INT (N/16):LET LB = N-H
     B*16
3020 LET B$ = X$(HB + 1) + X$(LB + 1)
3030 RETURN
3100 REM*****S/R D-H******
3110 IF NM<256 THEN LET N = NM GOS
     UB 3000:LET H$ = "00" + B$:RETURN
3120 LET HI = INT (NM/256):LET LO = N
     M-256*HI
3130 LET N = HI:GOSUB 3000:LET H$ =
     B$
3140 LT N = LO:GOSUB 3000:LET H$ =
     H$ + B$
3150 RETURN
4000 REM*****S/R H-D******
4050 LET RX = 1:LET DN = 0:LET HL = LE
     N(H$):IF (HL<1) OR (HL>4) THEN L
     ET DN = -1:RETURN
4100 FOR H = HL TO 1 STEP -1
4150 LET D$ = H$(H)
4200 LET V = CODE(D$) + C1*(D$> = "0"
     AND D$< = "9") + C2*(D$> = "A" AND D
     $< = "F")
4250 IF V>15 THEN LET DN = -1:LET
     H = 1:NEXT H:RETURN
4300 LET DN = DN + V*RX:LET RX = RX*16
4350 NEXT H:RETURN
4500 REM*****S/R FICTICIA*****
4550 RETURN
5000 REM*****S/R ALTER*****
5020 FOR P = 0 TO 1 STEP 0
5040 PRINT A$;:INPUT "NUEVO VALOR HEXA
     (X = SALIR)?";V$
5050 PRINT V$
5060 GOSUB 5340 :REM CHK&OBS
5080 NEXT P:RETURN
5200 REM*S/R PRUEBA-AJUSTE**
5220 IF A$ = "X" THEN LET P = 2:RETURN
5240 LET LL = LEN(A$):IF LL>4 THEN
     RETURN
5260 LET H$ = A$:GOSUB 4000
5280 IF DN> = 0 THEN LET P = 2:LET N
     M = DN
5300 LET A$ = A$ + " ":IF LL<4 THEN
     LET A$ = "0000" (TO 4-LL) + A$
5320 RETURN
5340 REM**S/R PRUEBA-OBSERV*
5360 IF V$ = "X" THEN LET P = 2:RETURN
5380 LET H$ = V$:GOSUB 4000
5400 IF (DN<0) OR (DN>255) THEN
     RETURN
5420 POKE NM,DN
5440 LET NM = NM + 1:IF NM>65535 THEN
     LET P = 2:RETURN
5460 GOSUB 3100 :REM D-H S/R
5480 LET A$ = H$ + "  ":RETURN
5500 REM****S/R DISPLAY*******
5520 FOR P = 0 TO 1 STEP 0
5540 INPUT "NO. DEC. DE BYTES (X = FUERA)"
     ;N$:IF N$ = "X" THEN LET P = 2:NE
     XT P:RETURN
5560 LET BN = VAL(N$):IF (BN>0) AN
     D (BN + NM < (65536) THEN LET P = 2
5580 NEXT P
5600 FOR B = NM TO (NM + BN-1) STEP
     4
5620 LET L$ = " ":LET NM = B:GOSUB 31
     00
5640 PRINT H$;TAB(6);
5660 FOR C = 0 TO 3
5680 LET N = PEEK(B + C):LET K$ = "."
5700 GOSUB 3000:REM S/R D-H
5720 PRINT TAB(6 + 4*C);B$;
5740 IF N = 0 THEN LET K$ = "■"
5760 IF (N>31) AND (N<128) THEN
     LET K$ = CHR$(N)
5780 LET L$ = L$ + K$
5800 NEXT C
5820 PRINT TAB(26);L$
5840 NEXT B:RETURN
6000 REM*****S/R EXECUTE****
6050 RANDOMIZE USR(NM):RETURN
6500 REM*****S/R EXIT******
6550 PRINT TAB(5);"■■■■FIN DE
     PROGRAM■■■■"
6600 LET Z = 2:RETURN
```

BBC Micro

```
39 REM**********************************
40 REM*        HCAC MONITOR 1           *
41 REM*        ---------BBC----------   *
42 REM*      CAMBIE LA VERSION          *
43 REM*    SPECTRUM DE ESTE MODO:       *
44 REM*                                 *
45 REM*    REEMPLACE CODE(POR ASC(      *
47 REM*                                 *
50 REM*    AÑADA, CAMBIE O BORRE        *
51 REM*      SEGUN INDICACION:          *
52 REM*                                 *
53 REM**********************************
60 *TV 255
70 MODE 7
200 PRINT CHP$(147);CHR$(142)
600 ON CM GOSUB 5000,5500,6000,6500
1050 LT = 4;DIM C$(LT),O$(LT)
1150 C$(1) = "A":C$(2) = "D":C$(3) = "G":C$(3)
     = "Q":C1 = 48:C2 = ASC(C$(1)) - 10
2190 ------------------ BORRE ------------------
3020 B$ = MID$(X$,HB + 1,1) + MID$(X$,LB + 1,1)
4150 D$ = MID$ (H$,H,1)
4500 ------------------ BORRE ------------------
4550 ------------------ BORRE ------------------
5050 ------------------ BORRE ------------------
5300 A$ = A$ + "  ":IF LL<4 THEN A$ = LEFT$("0
     000",4-LL) + A$
5420 ?(INM) = DN
5680 N = ? (B + C):K$ = "."
5740 IF N = 13 THEN K$ = CHR$ (255)
6050 CALL NM:RETURN
6600 Z = 1:RETURN
```

Commodore 64

```
49 REM ****************************************
50 REM*             HCAC MONITOR 1             *
51 REM* ---------------- CBM ----------------  *
52 REM*           CAMBIE LA VERSION            *
53 REM*        SPECTRUM DE ESTE MODO:          *
54 REM*                                        *
55 REM*            CAMBIE SIEMPRE              *
56 REM*       "LET P = 2" POR "P = 1"          *
57 REM*                                        *
58 REEMPLACE CODE (POR ASC)                    *
59 REM*                                        *
60 REM*          Y CAMBIE O BORRE              *
61 REM*          SEGUN INDICACION              *
62 REM*                                        *
63 REM ****************************************
200 PRINT CHR$(147);CHR$(142)
600 ON CM GOSUB 5000,5500,6000,6500
1050 LT = 4:DIM C$(LT),O$(LT)
1150 C$(1) = "A":C$(2) = "D":C$(3) = "G":C$(3)
     = "Q":C1 = 48:C2 = ASC(C$(1))-10
2190 ------------------ BORRE ------------------
3020 B$ = MID$(X$,HB + 1,1) + MID$(X$,LB + 1,1)
4150 D$ = MID$(H$,H,1)
4500 ------------------ BORRE ------------------
4550 ------------------ BORRE ------------------
5050 ------------------ BORRE ------------------
5300 A$ = A$ + "  ":IF LL<4 THEN A$ = LEFT$("0
     000",4-LL) + A$
5740 IF N = 0 THEN K$ = CHR$(122)
6050 SYS(NM):RETURN
6600 Z = 1:RETURN
```

Este programa le permitirá visualizar el contenido de la memoria, alterar el contenido de la misma y ejecutar un programa en lenguaje máquina almacenado



Historia de un éxito

Tras recorrer un azaroso camino, las cifras de ventas de Commodore son la envidia de los fabricantes de ordenadores de todo el mundo

El secreto del éxito de Commodore radica en una operación internacional de fabricación perfectamente organizada, representada en Gran Bretaña por una fábrica moderna —construida con subvenciones del gobierno— en la ciudad siderúrgica de Corby. Dicha fábrica se encuentra en proceso de expansión. En el presente emplea alrededor de 250 personas y la producción media asciende a 5 000 ordenadores diarios. Como todas las fábricas de Commodore, se beneficia del suministro asegurado de componentes semiconductores por parte de las propias fábricas de la casa matriz. Commodore es una enorme corporación y, dado su volumen de producción, puede llegar a un trato ajustado con los proveedores del exterior: en algunos casos, sólo paga la mitad de lo que desembolsan sus competidores por chips insustituibles.

La firme posición de Commodore se debe en gran medida al éxito del CBM PET (*Personal Electronic Transactor*: gestor electrónico personal). Su diseño fue esencialmente de Chuck Peddle, quien tomó tres decisiones importantes. La primera consistió en construir la máquina con el procesador 6502; la segunda fue equiparla con BASIC Microsoft, y la tercera, proporcionar un editor de pantalla completo, que permitió que la máquina fuera mucho más fácil de utilizar que los tableros únicos con que los entusiastas de la microelectrónica habían estado jugando desde la aparición del microprocesador, a mediados de la década de los setenta. Hasta hoy, el IBM PC, considerado de una generación más reciente que el PET, carece de un editor de pantalla completo como pieza estándar.

Commodore estaba en condiciones de construir dicha máquina pues ya era propietario de MOS Technology, que tenía los derechos para producir el microprocesador 6502. Este elemento resultó valiosísimo: los dos competidores del PET (el Apple II y el Tandy TRS-80) utilizaban la CPU 6502, por lo que Commodore estaba en condiciones de controlar su producción. De todos modos, el nacimiento del PET planteó problemas. En contra de los deseos de Peddle, Jack Tramiel insistió en que los componentes de memoria del PET también provinieran de MOS Technology. Ello condujo a una discusión muy aireada por la prensa entre ambos hombres y al brusco abandono de la empresa por parte de Peddle.

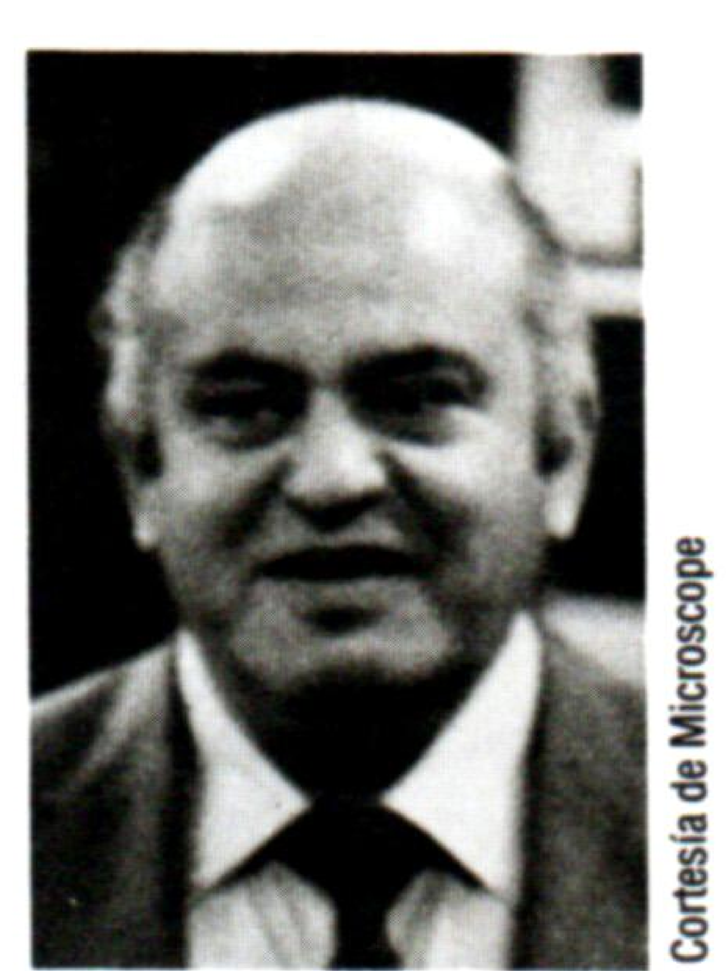
Ahora el PET es una máquina prestigiosa. Originalmente albergado en una caja de acero prensado (tal como solía construirse el mobiliario de oficina), en la actualidad se presenta recubierto de plástico a fin de modernizar su aspecto. Con esta última apariencia y con un potencial para manejar un almacén de discos de 22 megabytes, se le conoce como la serie 8000. Pese a su antigüedad, sigue vendiéndose extraordinariamente bien a los clientes más conservadores, que se sienten cómodos con él y no encuentran motivos para cambiar su software por la nueva generación de ordenadores de oficina basados en la CPU 8088. Sus proveedores incluso sostienen que, para su propio desconcierto, aún compite con el IBM PC y el Apricot de ACT.

Ello podría ser el resultado de un accidente histórico. Junto con Apple y Tandy, Commodore fue de las primeras empresas en introducir un ordenador personal asequible y fácilmente utilizable. Y en cuanto conquistó clientes, Commodore hizo todo lo posible por no perderlos. Tanto por conservadurismo como por deseo de limitar los costes de producción, Commodore jamás se preocupó por revisar las especificaciones de su máquina para principiantes. La mayoría del software original del PET aún se utiliza en las máquinas de hoy y el BASIC Microsoft versión 2.0 aún es prácticamente igual al adoptado por Peddle.

Lamentablemente para el aficionado más avanzado, el Microsoft 2.0 se desarrolló en una época en que los símbolos gráficos y el sonido constituían un lujo para un ordenador barato. Si bien los recientes ordenadores para aficionados de Commodore están perfectamente provistos de esas configuraciones, el BASIC carece de las instrucciones necesarias y requiere un amplio y laborioso uso de POKE para direccionar posiciones de memoria específicas. Pero esto no es un problema con el software de cartucho escrito de antemano, del que Commodore ofrece una amplia variedad.

Debido, al menos parcialmente, a su provisión constante de componentes, Commodore logró superar la tormenta que hundió a sus competidores en el mercado de juegos. Texas y Mattel se retiraron totalmente del mercado de ordenadores personales y Atari prácticamente ha estado ausente en estos dos últimos años. Gracias al suministro asegurado de piezas baratas, Commodore logró deslizar los precios del software de ordenadores y de cartuchos muy por debajo del precio con el que otros fabricantes podían competir, sin por ello dejar de obtener beneficios. En cierto momento, los cartuchos de Commodore costaban un tercio de lo que costaban los de sus competidores. Una planta moderna y automatizada y el acceso a recambios baratos fueron resultado de la experiencia de fabricación durante dos décadas. Se afirma que el precio de fábrica de un Commodore 64 es sólo la décima parte de su precio de venta.

Commodore no ha alcanzado esta posición de predominio por pura casualidad. Ha tenido sus altibajos y, al menos dos veces, estuvo al borde de la



Cortesía de Microscope

Jack Tramiel
La fuerza impulsora de Commodore ha sido su presidente, Jack Tramiel. Su sagaz capacidad comercial se echará de menos ahora que ha dejado de dedicarse plenamente a la empresa



Cortesía de VNU

Chuck Peddle
Chuck Peddle es el hombre que se encuentra tras el diseño del PET y el chip 6502 que contiene. Creó su propia empresa para producir la máquina de oficina Sirius, de 16 bits

bancarrota. Jack Tramiel, que llegó a Estados Unidos después de la segunda guerra mundial, siendo adolescente, en calidad de refugiado polaco de Auschwitz, fundó la CBM (*Commodore Business Machines*) en 1955. Estableció su sede en Toronto (Canadá), donde la nueva sociedad inició modestamente su producción montando máquinas de escribir con licencia checoslovaca. Se dice que Tramiel eligió el nombre de la empresa precisamente por su parecido con IBM.

En 1975, luego de dos décadas de comerciar con productos para oficinas, la empresa fue abatida por la feroz guerra de las calculadoras, que por el momento ganaron los japoneses. Pero Tramiel, tan temido como respetado por sus métodos comerciales, era un superviviente. Reparó en la existencia de un mercado potencial para un ordenador personal, en 1976 contrató a Chuck Peddle y en menos de una década vio cómo se multiplicaba por cincuenta el valor de la empresa.

Tramiel ha hecho de Commodore un monstruo del mercado y de la fabricación pero, si alguna debilidad tiene, ésta radica en la creación de nuevos productos. La filosofía de la empresa es "vendemos a las masas, no a las clases", y Tramiel está convencido de que el cliente siempre comprará lo que haga más rentable su dinero. Es posible que los requisitos de la fabricación barata a gran escala vayan contra la incorporación de la tecnología más avanzada. A fines de 1982 un alto porcentaje del pequeño equipo de investigación y desarrollo de Commodore abandonó la empresa en una retirada en masa y desde entonces ésta se ha dedicado a comprar los frutos de las investigaciones de otras empresas. Ha firmado acuerdos de fabricación con empresas del Lejano Oriente para producir unidades de disco y ha celebrado conversaciones con firmas como Sony a fin de adquirir costosas tecnologías de la "quinta generación" como reconocimiento de voz, robots domésticos y sofisticados dispositivos de almacenamiento. Incluso ha abordado a Paul Johnson, diseñador creativo de Oric, para tratar de que le diseñe un chip ULA destinado a su nueva serie de ordenadores personales.

En 1984 Commodore se muestra tan confiada como siempre y sigue haciendo una virtud del bajo coste y la sencillez. Ha presentado dos nuevos ordenadores de uso doméstico basados en los nuevos procesadores 7501 y conocidos como el 264 y el V364. Este último tiene un sintetizador de voz y un vocabulario incorporado de 250 palabras. En línea con las tendencias actuales, como opción podrá disponerse de software para tratamiento de textos, hoja electrónica y símbolos gráficos.

Hitos de Commodore



1977
El **Commodore PET** original fue el primer ordenador personal para el mercado de masas. Después de múltiples modernizaciones, sigue vendiéndose bien

1984
El **SX64** es una versión actualizada del 64, con caja portátil, pantalla en color y unidad de disco

1982
El **CBM 700** es el reemplazo de las máquinas 8032 que promete poner las máquinas de oficina de Commodore a la altura de los micros más modernos

1979
El **Vic-20** fue el primer ordenador de uso doméstico barato de Commodore y, pese a sus limitaciones y a una fuerte competencia, sigue siendo muy popular

1981
El **CBM 8032** añadió una capacidad de 80 columnas al PET. Ello permite usar avanzados programas de oficina

1983
El **Commodore 64** ha mejorado las limitaciones del Vic al contar con pantalla de 40 columnas y memoria de 64 K

1980
El **SuperPET** (o CBM 9000) fue un intento de producir una versión de oficina del PET espectacularmente modificada

Ian McKinnell, Chris Stevens y cortesía de Commodore

Editorial Delta, S.A.



10030

9 788485 822836